GENERAL ATOMIC DIVISION
GENERAL DYNAMICS CORPORATION

John Jay Hopkins Laboratory for Pure and Applied Science
P.O. Box 608, San Diego 12, California

GAMD-5580-Revised
Category A

Copy No.

OIL

A CONTINUOUS TWO-DIMENSIONAL EULERIAN HYDRODYNAMIC CODE

Work done by:

Project Staff

Report written by: [illegible]

W. [illegible] Johnson

This document, which was prepared primarily for internal use at General Atomic, may contain preliminary or incomplete data. It is informal and is subject to revision or correction; it does not, therefore, represent a final report.

Advanced Research Projects Agency
ARPA Order No. 71-62
Ballistic Research Laboratories
Army Contract DA-04-495-AMC-116(x)

Project No. [illegible]
January [illegible], 1965

FOREWORD

The GIL computer code described herein is as it existed on October 26, 1964. The code has been in continuous development for three years and in its presented form has been applied successfully by General Atomic to the kind of problems discussed in this report. However, the development and improvement in both the physics and mathematics of the code are being continued, so that duplication of results (or even close agreement) between problems run with the code as published and the code as it existed either before or after this time is not necessarily to be expected.

General Atomic has exercised due care in preparation, but does not warrant the merchantability, accuracy, and completeness of the code or of its description contained herein. The complexity of this kind of program precludes any guarantee to that effect. Therefore, any user must make his own determination of the suitability of the code for any specific use and of the validity of the information produced by use of the code.

CONTENTS

1. INTRODUCTION AND ACKNOWLEDGEMENTS

The OIL code is very closely related to familiar particle-in-cell codes[1,2,3] and has been developed by modification of the General Atomic particle-in-cell code named SHELL. The basic difference lies in the method of the mass transport, the OIL scheme being a continuous mass transport, while the SHELL scheme is discrete. The initial work on a continuous version of SHELL was undertaken several years ago by B. E. Freeman and the author. Since early 1963 the development of the continuous Eulerian has been continued by J. M. Walsh and the author.

The author is deeply appreciative of the work and consultation given by B. E. Freeman and J. M. Walsh. The assistance of D. R. Yates and R. H. Fisher in the automatic plotting routines used for the OIL code is also much appreciated.

Since its original use for the hypervelocity impact calculations[4], the OIL code has been successfully adapted to several other high energy fluid dynamic applications.

Detailed descriptions of various problems, especially results for hypervelocity impact, are given in Ref. 4.

2. CLAM

2.1. General Description

CLAM is the generator code for the OIL code, and is used to generate initial values of the variables for each cell in the grid. (An exception arises for certain simple initial conditions, described in Section 3.2, where it is possible to bypass CLAM and use instead a more economical routine called SETUP.)

The function of CLAM is illustrated by a simple example (Fig. 1). We wish to generate the following grid: A right circular cylinder of density 1. g/cm^3, radius 24 cm, and height 12 cm is located at position z = 20 cm along the axis. A projectile (right cylinder) of density 1. g/cm^3, radius 6 cm, and height 12 cm, is located at positions 8-20 along the axis. The projectile has an initial axial velocity of 1 x 10^6 cm/sec. The cells are 1 cm on a side.

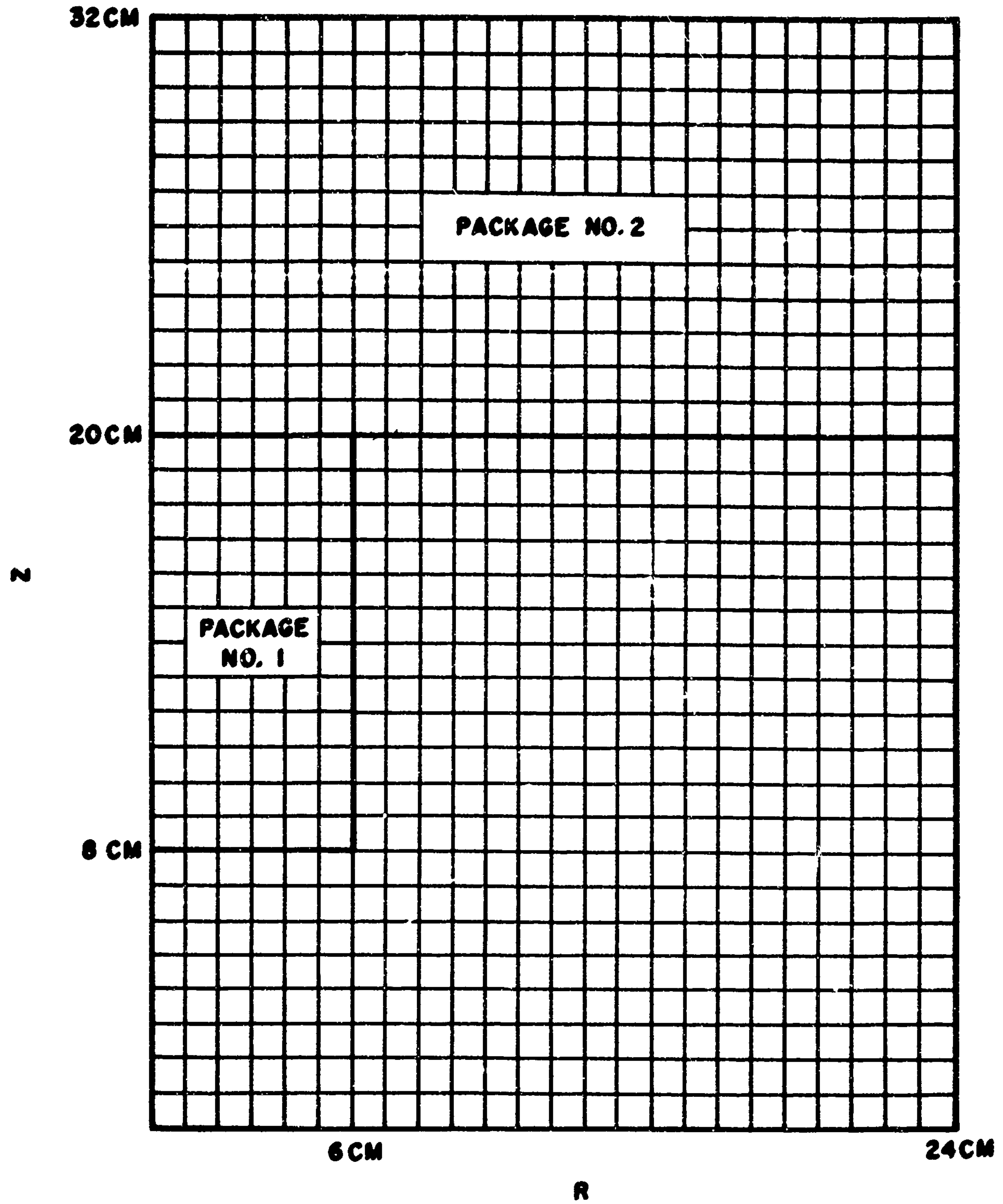


Fig. 1

(CLAM INPUT)

```
        GAS  GAMMA = 1.5  RHO TARGET = RHO PELLET =1.
1.          24.         32.         0.          2.          0.          0.          07
0024        1.
1132        1.
2.          3.          80.
11    01    0.          0.          1.
4     1     0.          6.          8.          20.
51          1.0         0.          0.          0.          0.          0.
52          0.          0.          0.          0.          0.          0.
53          .01         0.          0.          0.          0.          0.
11    01    0.          0.          1.
4     1     0.          24.         20.         32.
53          0.          0.          0.          0.          0.          0.
2
```

GAS   GAMMA = 1.5 RHO TARGET= RHO PELLET = 1.

PROB. NO. 1.000   I=24   J=30

X(I) I=0,24

0.0	1.0	2.0	3.0	4.0
5.0	6.0	7.0	8.0	9.0
10.0	11.0	12.0	13.0	14.0
15.0	16.0	17.0	18.0	19.0
20.0	21.0	22.0	23.0	24.0

Y(J) J=0,32

0.0	1.0	2.0	3.0	4.0
5.0	6.0	7.0	8.0	9.0
10.0	11.0	12.0	13.0	14.0
15.0	16.0	17.0	18.0	19.0
20.0	21.0	22.0	23.0	24.0
25.0	26.0	27.0	28.0	29.0
30.0	31.	32.0		

DX(I) I=1,24

1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	

DY(J) J=1,32.

1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0
1.0	1.0			

AREA(I) I=1,48

3.141593	9.424778	15.707963	21.991149	28.274334
34.557519	40.840705	47.123890	53.407076	59.690261
65.973447	72.256632	78.539817	84.823003	91.106188
97.389374	103.672559	109.955745	116.238930	122.522116
128.805300	135.088486	141.371672	147.654856	

PACKAGE NO. 1 1 PARTICLE /CELL

	A1	A2	A3	A4	A5	A6
DENSITY	1.	0.	0.	0.	0.	0.
ENERGY	0.	0.	0.	0.	0.	0.
VELOCITY	0.01	0.	0.	0.	0.	0.

RECTANGLE---GEN---- 0. 6. 8. 20.
I(1)= 1 J(1)= 8 I(N)= 7 J(N)= 21

72 (X)PARTICLES PE=6.785839-02 PM=1.357168+03

PACKAGE NO. 2 1 PARTICLE /CELL

	A1	A2	A3	A4	A5	A6
VELOCITY	0.	0.	0.	0.	0.	0.

RECTANGLE--- GEN--- 0. 24. 20. 32.
I(1)= 1 J(1)=20 I(N)=24 J(N)=32

288 (X) PARTICLES PE=0. PM =2.171467+04

THE = 6.785838399-2 E = 6.785839237-2

M. = 0. MX = 2.30718+04 M.+ MX = 2.30718+04

PARTICLES --- 0 DOT 360 X 360 TOTAL

TAPE DUMP AT TIME 0. CYCLE 0.

The coordinates $x(i)$ and $y(j)$ of each cell are calculated from the DX's and DY's that are loaded. In addition, the area term $(TAU_{(i)}) = (x^2_{(i)} - x^2_{(i-1)})\pi$ are calculated.

The next step is to process the packages (see Section 2.2); there are two in this example. Package Number (1) is the projectile and Package Number (2) is the target.

We specify the coordinates of each package. In addition, we specify the number of particles per cell in each package. We assign a density, the two velocity components, and specific internal energy for the package in question. These may be constants or functions of both z and r.

After specifying the coordinates of the package, particles (N, where $N^{\frac{1}{2}}$ may range from 1 to 20) are placed in each cell. Each of the particles is then assigned a density, the two velocity components and specific internal energy. The cell is divided into N equal parts, and the N particles placed at the center of these areas. The mass of each particle is then the density (prescribed along with the package input) times the volume of the small subdivision cell of the cell (K). The mass of cell (K) then is the sum of the N particle masses, and both momenta components are calculated as the sum of the individual momenta components of each particle. The internal energy of each cell (K) is the specific internal energy times the particle mass summed over all N particles.

In the output routine these cell quantities are then converted to the two velocity components and the specific internal energy.

Particles are created, whether these data are for an OIL or a SHELL run, for the purpose of describing the many possible geometries and possible energy, velocity, and density distributions. If this is an OIL run, the particles are not written on a tape as they would be if this were a SHELL run. One continues to process each cell (K) within the given package and the N particles in cell (K).

In this example, it is sufficient to specify only one particle per cell, since the package boundaries coincide with the cell boundaries and the density, velocities, and internal energy are constant.

After processing all packages, one then writes this information about the grid and the cells on a tape that will be the starting conditions for the OIL or SHELL code (see Section 2.3.)

The output from CLAM, excluding the long print, is shown on the listings on pages 3 and 4.

2.2. Input Description for CLAM

CLAM and OIL are written in cylindrical coordinates with axial symmetry. In the following discussion and description, X refers to the coordinate R and Y to the coordinate Z. An asterisk before the word signifies that the data are in floating point; otherwise they are fixed point.

Card No.	Column No.	Description
1		Header card, any BCD information in col. 2-72.
2	* 1-10	Contain the problem number. If less than zero, this will be a particle run using SHELL; if greater than zero, this will be a continuous run using OIL.
	* 11-20	IMAX, the number of cells in the X-direction (maximum of 50.)
	* 21-30	JMAX, the number of cells in the Y-direction (maximum of 100)
	* 31-40	Blank
	* 41-50	2
	* 51-60	S8 (not used as input in CLAM)
	* 61-70	S9 (not used in CLAM)
	* 71-72	N7 (N7 = binary tape number)
3		(2 cards is the minimum)
	1	A(1) indicates that this is the last DX or DY card to be read in. A(0) indicates that there will be more DX or DY data cards.
	2	A(0) indicates that we are loading DX data A(1) indicates DY data.
	3-4	Indicate the number of zones that will have the same DX or DY values that appear in columns 11-20.
	5-6	Indicate the number of zones that will have the same DX or DY values that appear in columns 21-30.

Card No.	Column No.	Description
3	7-8	Indicate the number of zones that will have the same DX or DY values that appear in columns 31-40.
	9-10	Indicate the number of zones that will have the same DX or DY values that appear in columns 41-50.
	* 11-20	The value of DX or DY
	* 21-30	The value of DX or DY
	* 31-40	The value of DX or DY
	* 41-50	The value of DX or DY
4	* 1-10	N1 = the tape number of one of the scratch tapes to be used in CLAM and OIL
	* 11-20	N2 = the other tape number for the scratch tape. OIL requires two scratch tapes if using the particle transport.
	* 21-30	N4 = maximum number of particles plus one per tape record, that CLAM will generate (maximum value = 130)
	* 31-40	Switch (not used)

Now we begin loading the necessary data to generate a package. The maximum number of packages that may be generated is 72; to increase the maximum requires changing the dimension statements.

1	1	Load a 1 here
	2	A(1) implies that x material will be generated in this package. A(0) implies that dot material will be generated.
	5-7	(N^2), the number of particles per cell to be generated, where $1 \leq N \leq 20$.
	* 11-20	YC = Y coordinate for the origin of the radius vector used in the density, energy, and velocity fits.
	* 21-30	XC = X coordinate for the origin of the radius vector used in the density, energy, and velocity fits.
	* 31-40	TEMP(3) is loaded into S8 (S8 contains the number (1-6) of the fit for the density, internal energy, and velocity that will be assigned to each particle as it is generated in this package.
	* 41-70	Blank.

Following the first card of each package are five other types of cards:

(1) Generate (geometry of package)

(2) Delete card (if needed); there may be more than one

(3) A density card (only one per package)

(4) An energy card (only one per package)

(5) A velocity card (only one per package)

For card (1) above, CLAM has the following geometric options for generating or deleting:

1. A rectangle - a 4 in column 1

   a (1) in column 7 means generate this rectangle

   a (0) in column 7 means delete this rectangle

   * 11-20 X1 = the left R coordinate of the rectangle

   * 21-30 X2 = the right R coordinate of the rectangle

   * 31-40 Y1 = the lower Z coordinate of the rectangle

   * 41-50 Y2 = the upper Z coordinate of the rectangle

2. A triangle - a 6 in column 1

   a (1) in column 7 means generate this triangle

   a (0) in column 7 means delete this triangle.

   * 11-20 X1)

   * 21-30 Y1)

   * 31-40 X2)--Note: Vertices (1-3) can be in any order

   * 41-50 Y2)

   * 51-60 X3)

   * 61-70 Y3)

3. An ellipse or circle - a 41 in column (1-2)

   a (1) in column 7 means generate this ellipse or circle

   a (0) in column 7 means delete this ellipse or circle

* 11-20 The semi-axis in the X-direction if an ellipse or th the radii if for a circle
* 21-30 The semi-axis in the Y-direction if an ellipse or zero if for a circle
* 31-40 The X-coordinate of the center of ellipse or circle
* 41-50 The Y-coordinate of the center of ellipse or circle.

4. A perturbed ellipse - a 61 in columns (1-2)

   a (1) in column 7 means generate this perturbed ellipse.

   a (0) in column 7 means delete this perturbed ellipse.

* 11-20 Semi-axis in the X-direction
* 21-30 Semi-axis in the Y-direction
* 31-40 0.
* 41-50 Y-coordinate of center of perturbed ellipse
* 51-60 X-coordinate of point of perturbation
* 61-70 Y-coordinate of point of perturbation.

Following the geometry cards are the following:

Density card - a 51 in columns (1-2)

Energy card - a 52 in columns (1-2)

Velocity card - a 53 in columns (1-2)

Note: If in this package, the ρ or I or velocity will remain the same as the previous package, then a 51, 52 or 53 card is not required.

* 11-20)
* 21-30)
* 31-40)-Contains the values to be used in the analytical
* 41-50) expressions for the density, energy, and velocities.
* 51-60)
* 61-70)

This data is then loaded into the following arrays:

TABR(1-6) for density

TABi(1-6) for internal energy

TABUV(1-6) for velocity

Finally, a 2 in column 1 signifies the completion of loading all input cards for the CLAM code.

There are six subroutines (FIT1 - FIT6) used for computing ρ, I, U, and V. The standard input to these subroutines is as follows:

TY = Y coordinate of particle N

TX = X coordinate of particle N

The modified coordinates TTY and TTX are computed as follows:

TTY = Y coordinate = TY - YC (relative to YC)

TTX = Y coordinate = TX - XC (relative to XC)

The standard output from the subroutines is as follows:

WSR = ρ (density) of particle N

WSi = I (specific internal energy) of particle N

WSU = U (radial velocity component) of particle N

WSV = V (axial velocity component) of particle N.

Below are the equations, or analytical fits, for the six subroutines. Any or all may be changed. Each equation is followed by the FORTRAN mnemonic.

1. <u>FIT 1</u>

$$R = (X^2 + Y^2)^{\frac{1}{2}}$$

$$WS = (TTX^2 + TTY^2)^{\frac{1}{2}}$$

$$\rho = A + B\ (Y - C)$$

WSR = TABR(1) + TABR(2) * (TTY - TABR(3))

$$I = A + B\ (Y-C)$$

WSi = TABi(1) + TABi(2) * (TTY - TABi(3))

$$U = 0$$

WSU = 0

$$V = A + B\ (Y - C)$$

WSV = TABUV(1) + TABUV(2) * (TTY - TABUV(3))

2. <u>FIT 2</u>

$$R = (X^2 + Y^2)^{\frac{1}{2}}$$

$$WS = (TTX^2 + TTY^2)^{\frac{1}{2}}$$

$$\rho = \left(\frac{X-A}{B}\right)^2 + \left(\frac{Y-C}{D}\right)^2$$

$$WSR = \left(\frac{TTX - TABR(1)}{TABR(2)}\right)^2 + \left(\frac{TTY - TABR(3)}{TABR(4)}\right)^2$$

$$I = A + BX + CX^2 + DY + EY^2$$

$$WSi = TABi(1) + TABi(2) * TTX + TABi(3) * TTX^2 + TABi(4) * TTY + TABi(5) * TTY^2$$

$$U = C + D * Y$$

$$WSU = TABUV(3) + TABUV(4) * TTY$$

$$V = A + B * Y$$

$$WSB = TABUV(1) + TABUV(2) * TTY$$

3. <u>FIT 3</u>

$$R = (X^2 + Y^2)^{\frac{1}{2}}$$

$$WS = (TTX^2 + TTY^2)^{\frac{1}{2}}$$

$$\rho = A + BR + CR^2$$

$$WSR = TABR(1) + TABR(2) * WS + TABR(3) * WS^2$$

$$I = A + BR + CR^2$$

$$WSi = TABi(1) + TABi(2) * WS + TABi(3) * WS^2$$

$$U = \frac{X}{R}\left(\frac{A + BR + CR^2}{D + ER + FR^2}\right)$$

$$WSU = \frac{TTX}{WS}\left(\frac{TABUV(1) + TABUV(2) * WS + TABUV(3) * WS^2}{TABUV(4) + TABUV(5) * WS + TABUV(6) * WS^2}\right)$$

$$V = \frac{Y}{R}\left(\frac{A + BR + CR^2}{D + ER + FR^2}\right)$$

$$WSV = \frac{TTY}{WS}\left(\frac{TABUV(1) + TABUV(2) * WS + TABUV(3) * WS^2}{TABUV(4) + TABUV(5) * WS + TABUV(6) * WS^2}\right)$$

Fits 4, 5, and 6 are dummy routines. Although particles are not used in the SHELL code if it is a continuous run (problem number greater than zero), the use of particles in CLAM provides the method for assigning mass, energy, to each cell.

2.3. OUTPUT FROM CLAM

The output from the CLAM code is the entire Z block (defined below), all the cell quantities (the two velocity components, the mass and internal energy), and the cell dimensions and areas. In the case where it is a particle run, the particles (their two coordinates and mass) and the i and j of the cell where the particle is located are also put onto the binary tape.

The normal system of units are the centimeter-gram-shake, where the units of energy are jerks/g and the pressure in units of jerks/cm^3 (1 jerk = 10^{16} ergs and 1 shake = 10^{-8} sec).

The Z block or array contains the number of cells, the number of zones in both directions, and other necessary information to start the OIL or SHELL calculation. Below is a complete list of generated data from CLAM that is written on the binary output tape.

Z	Equiv.	Units	Description
1	PROB	--	Equals problem number, input to CLAM
2	Cycle	--	Equals cycle number = 0
3	DT	shake	Set to 0 by CLAM
4	Prints	--	
5	Print-L	--	
6	DUMPT 7	--	
7	C Stop	--	
8	PIDY	--	Equals π = 3.1415927
9	TMZ	grams	Total mass (x) generated by CLAM
10	GAM	--	Set to 0 by CLAM
11	GAMD	--	Set to 0 by CLAM
12	GAMX	--	Set to 0 by CLAM
13	ETH	jerk	Total energy in system
14	FFA	--	Set to 0 by CLAM
15	FFB	--	Set to 0 by CLAM
16	TMDZ	grams	Total mass (·) generated by CLAM; this version of CLAM does not generate (·)
17	TMXZ	grams	Total mass (x) generated by CLAM
18	XMAX	cm	= X(IMAX)

Z	Equiv.	Units	Description
19	TXMAX	cm	= 2 XMAX
20	TYMAX	cm	= 2. Y MAX (note Y MAX is not in Z block)
21	AMDM	grams	= minimum mass/2. of the dot particles
22	AMXM	grams	= minimum mass/2. of the X particles
23	DNN	--	Set to 0. by CLAM
24	DMIN	--	Set to 0. by CLAM
25	FEF	--	Set to 0. by CLAM
26	DTNA	--	Set to 0. by CLAM
27	CVIS	--	Set to 0. by CLAM
28	NPR	--	Set equal to 6 in CLAM
29	NPRi	--	CLAM sets NPRi = N4 (check definition of N4(Z(54))
30	NC	--	Fixed value of cycle number, set to 0. by CLAM
31	NPC	--	Used as indices in CLAM
32	NRC	--	Used as indices in CLAM
33	iMAX	--	Input to CLAM = maximum number of zones in X direction for this run
34	iMAX	--	Equal iMAX + 1
35	jMAX	--	Input to CLAM = maximum number of zones in Y direction for this run
36	jMAXA	--	= jMAX + 1
37	KMAX	--	= (iMAX)(jMAX) + 1
38	KMAXA	--	= KMAX + 1
39	NMAX	--	= total number of particles + 1 that CLAM has generated
40	ND	--	= total number of dot particles + 1 that CLAM has generated
41	KDT	--	Set to 0. by CLAM.
42	iXMAX	--	= iMAXA + 1
43	NOD	--	Used as index
44	NOPR	--	Set equal to N3 (Note definition of N3 (Z(53))
45	NiMAX	--	Set to 0. by CLAM
46	NjMAX	--	Set to 0. by CLAM

Z	Equiv.	Units	Description
47	i1		Set to 0. by CLAM
48	i2		Set to 0. by CLAM
49	i3		Set to 0. by CLAM
50	i4		Set to 0. by CLAM
51	N1	--	= scratch tape number
52	N2	--	= scratch tape number
53	N3	--	= number of particle records of length N[illegible] - [illegible] that CLAM has generated
54	N4	--	= number of particles + 1 to be stored on each particle tape record
55	N5	-	Set to 0. by CLAM
56	N6	--	= number of particles on the last particle tape record
57	N7	--	= binary tape designation number
58	N8		Set to 0. by CLAM
59	N9		
60	N10		
61	N11		
62	NRM		
63	TRAD		
64	XNRG		
65	SN		
66	DXN		
67	RADER		
68	RADET		
69	RADEB		
70	DTRAD		
71	REZFCT		
72	RSTOP		
73	SHELL		A counter that may be used to distinguish between codes
74	BBOUND		Set to 0. by CLAM
75	TOZONE		Set to 0. by CLAM

Z	Equiv.	Units	Description
76	ECK		Set to 0. by CLAM
77	SBOUND		
78	X1		
79	X2		
80	Y1		
81	Y2		
82	CABIN		
83	VISC		
84	T		
85	GMAX		
86	WSGD		
87	WSGX		
88	GMADR		
89	GMAXR		
90	S1		
91	S2		
92	S3		
93	S4		
94	S5		
95	S6		
96	S7		
97	S8	--	Used for storage of FIT number for each package in CLAM
98	S9		Set to 0. by CLAM
99	S10		Set to 0. by CLAM

Z(100) through Z(150) is also set to 0. by CLAM.

3. OIL

3.1. The Eulerian equations we wish to solve are the following:

$$\text{(A)}\quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0$$

$$\text{(B)}\quad \frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u}\vec{u}) = -\nabla P$$

$$\text{(C)}\quad \frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho E \vec{u}) = -\nabla \cdot (P\vec{u})$$

Equation (A) is the conservation of mass equation, (B) is the conservation of momentum, and (C) is the conservation of energy equation.

The second terms on the left side of Eqs. (B) and (C) are temporarily dropped. Their contributions are later approximated when we move particles or continuous mass across cell boundaries.

Rewriting equations (A), (B), and (C) in cylindrical coordinates with axis of symmetry results in Eqs. (1), (2), (3), and (4).

$$\frac{\partial \rho}{\partial t} = -\frac{\partial r\rho u}{r\partial r} - \frac{\partial \rho v}{\partial z} \qquad (1)$$

$$\rho\frac{\partial u}{\partial t} = -\frac{\partial P}{\partial r} \qquad (2)$$

$$\rho\frac{\partial v}{\partial t} = -\frac{\partial P}{\partial z} \qquad (3)$$

$$\rho\frac{\partial E}{\partial t} = -\frac{\partial rPu}{r\partial r} - \frac{\partial Pv}{\partial z} \qquad (4)$$

$$P = f(\rho, I) \quad \text{Equation of state} \qquad (5)$$

ρ = density of cell (K) in g/cm^3
r = r coordinate in cm.
z = z coordinate in cm.
u = radial component of velocity in cm/shake
v = axial component of velocity in cm/shake
P = material pressure in jerks/cm^3
E = total specific energy in jerks/g
I = specific internal energy in jerks/g

(1 jerk = 10^{16} ergs)

The five variables listed are all located at the center of the cell (Fig.2). For complete description of all quantities used, see Section 3.4 on List of Common for OIL."

Rewriting Eq. (4):

$$\rho \frac{\partial}{\partial t}\left[I + \tfrac{1}{2}(u^2 + v^2)\right] = -\frac{\partial rPu}{r\partial r} - \frac{\partial Pv}{\partial z}$$

or

$$\rho \frac{\partial I}{\partial t} + \rho u \frac{\partial u}{\partial t} + \rho v \frac{\partial v}{\partial t} = -\frac{P}{r}\frac{\partial ur}{\partial r} - u\frac{r\partial P}{r\partial r} - v\frac{\partial P}{\partial z} - P\frac{\partial v}{\partial z}$$

but

$$\rho \frac{\partial u}{\partial t} = -\frac{\partial P}{\partial r} \quad \text{and} \quad \rho \frac{\partial v}{\partial t} = -\frac{\partial P}{\partial z}$$

thus

$$\rho \frac{\partial I}{\partial t} = -P\left(\frac{\partial v}{\partial z} + \frac{1}{r}\frac{\partial ur}{\partial r}\right).$$

Rewriting the momentum equations as

$$\rho \frac{\partial (u^2)}{\partial t} = -2u\frac{\partial P}{\partial r} \quad \text{and} \quad \rho \frac{\partial (v^2)}{\partial t} = -2v\frac{\partial P}{\partial z}$$

the radial momentum equation becomes in difference form

$$\rho \frac{\partial (u^2)}{\partial t} = 2u_{i-\frac{1}{2},j-\frac{1}{2}}\, r_{i-\frac{1}{2}} \frac{[P_{i-3/2,j-\frac{1}{2}} - P_{i+\frac{1}{2},j-\frac{1}{2}}]}{\Delta(r_i^2)}$$

and the axial momentum equation becomes

$$\rho \frac{\partial (v^2)}{\partial t} = v_{i-\frac{1}{2},j-\frac{1}{2}} \frac{[P_{j-3/2,i-\frac{1}{2}} - P_{j+\frac{1}{2},i-\frac{1}{2}}]}{\Delta z_j}$$

and the change in specific internal energy becomes

$$\rho \frac{\partial I}{\partial t} = -P\left[\frac{v^{n+\frac{1}{2}}_{i-\frac{1}{2},j+\frac{1}{2}} - v^{n+\frac{1}{2}}_{i-\frac{1}{2},j-3/2}}{2\Delta z(j)} + \frac{r_{i+\frac{1}{2}}\, u^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}} - r_{i-3/2}\, u^{n+\frac{1}{2}}_{i-3/2,j-\frac{1}{2}}}{\Delta(r_i)^2}\right]$$

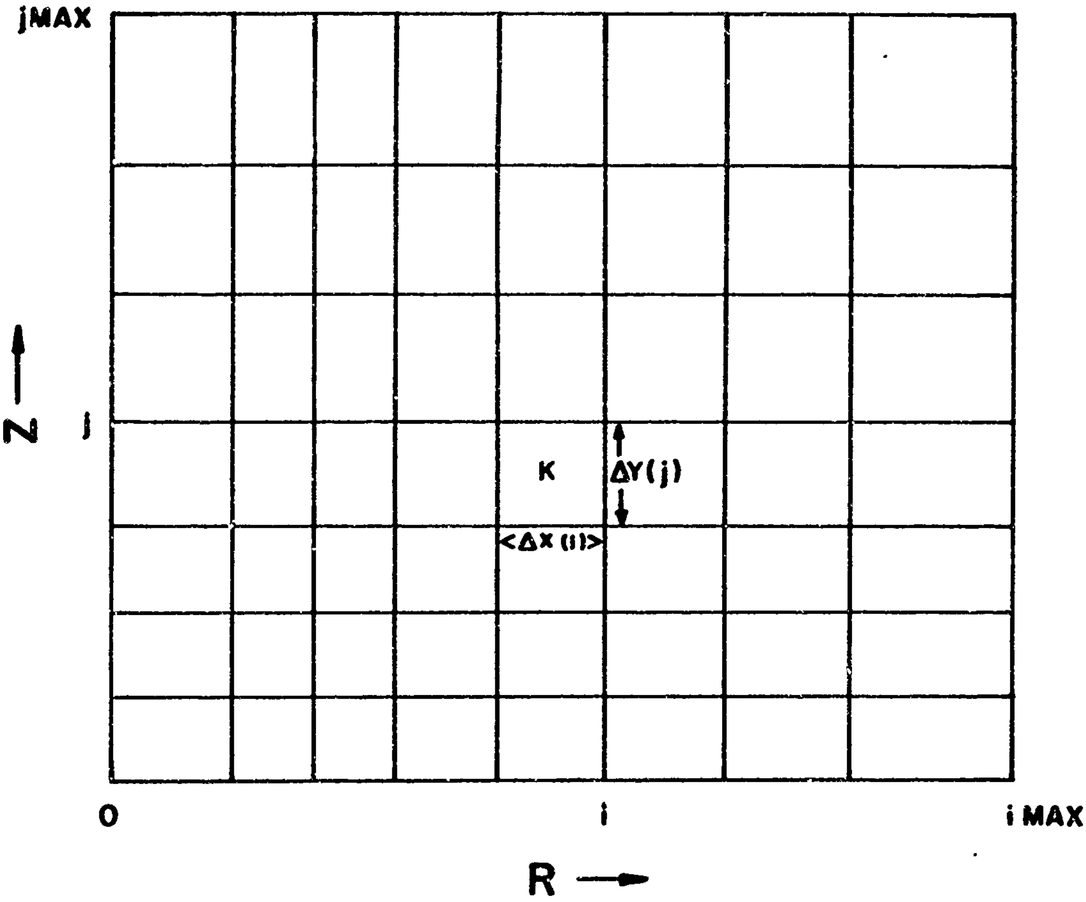
Defining the velocity on the right-hand side of the momentum equations at time $n+\frac{1}{2}$ results in

i IS THE RIGHT BOUNDARY

AND j IS THE TOP BOUNDARY OF THE CELL

$$X(i) = \sum_{i=1}^{i} \Delta X(i)$$

$$Y(j) = \sum_{j=1}^{j} \Delta Y(j)$$



THE AREA OF CELL (i, j) IN THE i DIRECTION $= 2\pi X(i) \; \Delta Y(j)$

THE AREA OF CELL (i, j) IN THE j DIRECTION $= \pi \left(X_{(i)}^2 - X_{(i-1)}^2 \right)$

K IS DEFINED AS (j-1) iMAX + i + 1, AND IS THE INDEX FOR CELL QUANTITIES

Fig. 2

$$\frac{\rho_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta t}\left[\left(u_{i-\frac{1}{2},j-\frac{1}{2}}^{n+1}\right)^2 - \left(u_{i-\frac{1}{2},j-\frac{1}{2}}^{n}\right)^2\right] = 2r_{i-\frac{1}{2}}\left(u_{i-\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}}\right)\frac{\left[P^n_{i-3/2,j-\frac{1}{2}} - P^n_{i+\frac{1}{2},j-\frac{1}{2}}\right]}{\Delta(r_i^2)}$$

and

$$\frac{\rho_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta t}\left[\left(v_{i-\frac{1}{2},j-\frac{1}{2}}^{n+1}\right)^2 - \left(v_{i-\frac{1}{2},j-\frac{1}{2}}^{n}\right)^2\right] = \left(v_{i-\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}}\right)\frac{\left[P^n_{j-3/2,i-\frac{1}{2}} - P^n_{j+\frac{1}{2},i-\frac{1}{2}}\right]}{\Delta z_j}$$

and

$$\frac{\rho_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta t}\left[I_{i-\frac{1}{2},j-\frac{1}{2}}^{n+1} - I_{i-\frac{1}{2},j-\frac{1}{2}}^{n}\right]$$

$$= -P_{i-\frac{1}{2},j-\frac{1}{2}}^{n}\left[\frac{v_{i-\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - v_{i-\frac{1}{2},j-3/2}^{n+\frac{1}{2}}}{2\Delta z_j} + \frac{r_{i+\frac{1}{2}}\, u_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} - r_{i-3/2}\, u_{i-3/2,j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta(r_i^2)}\right]$$

Defining:

$$u_{i-\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} = \frac{u_{i-\frac{1}{2},j-\frac{1}{2}}^{n+1} + u_{i-\frac{1}{2},j-\frac{1}{2}}^{n}}{2.}$$

$$v_{i-\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v_{i-\frac{1}{2},j-\frac{1}{2}}^{n+1} + v_{i-\frac{1}{2},j-\frac{1}{2}}^{n}}{2.}$$

and

$$P_i^n = \frac{P^n_{i+\frac{1}{2}} + P^n_{i-\frac{1}{2}}}{2.}$$

$$P_j^n = \frac{P^n_{j+\frac{1}{2}} + P^n_{j-\frac{1}{2}}}{2.}$$

Equation (2), the radial momentum equation, becomes

$$u^{n+1}_{(k)} - u^{n}_{(k)} = 2\Delta t\; r_{i-\frac{1}{2}}\; \pi DY_{(j)}\; \frac{(PL^n - PRR^n)}{AMX(k)}$$

where

$$PL^n = \frac{P^n_{(k)} + P^n_{(\text{cell to the left})}}{2.}$$

$$PRR^n = \frac{P^n_{(k)} + P^n_{(\text{cell to the right})}}{2.}$$

Equation (3), the axial momentum equation, becomes

$$v^{n+1}_{(k)} - v^{n}_{(k)} = \Delta t_{\pi} (r_i^2 - r_{i-1}^2) \frac{(P^n_{BLO} - P^n_{ABOVE})}{AMX(k)}$$

where

$$P^n_{BLO} = \frac{P^n_{(k)} + P^n_{(cell\ below)}}{2.}$$

$$P^n_{ABOVE} = \frac{P^n_{(k)} + P^n_{(cell\ above)}}{2.}$$

The energy equation (4) becomes

$$I^{n+1}_{(k)} - I^{n}_{(k)} = \frac{P^n_{(K)}\ \pi\Delta t}{AMX_{(k)}} \Big[\Big(\frac{VBLO^n + VBLO^{n+1} - VABOVE^n - VABOVE^{n+1}}{2.} \Big) x$$

$$(r_i^2 - r_{i-1}^2) + [DY(j)]\ (UL^n_j + UL^{n+1}_j - URR^n - URR^{n+1}) \Big]$$

where

$$VBLO = \frac{v_{(k)} + v_{(cell\ below)}}{2.}$$

$$VABOVE = \frac{v_{(k)} + v_{(cell\ above)}}{2.}$$

$$UL = \frac{u_{(k)}{}^{RC} + u_{(cell\ on\ the\ left)}{}^{RL}}{2.}$$

$$URR = \frac{u_{(k)}{}^{RC} + u_{(cell\ to\ the\ right)}{}^{RR}}{2.}$$

where

$$RC = r_{(i-\frac{1}{2})}$$

$$RR = r_{(i+\frac{1}{2})}$$

$$RL = r_{(i-3/2)}$$

The above equations conserve energy exactly, despite finite difference approximations. However, an adjustment at transmittive boundaries of the grid is necessary. This is a work term, which is also taken into account in ETH (the total energy of the system.) For the transmittive boundaries, the pressure gradient is zero and the velocity at the boundary interface is set equal to the velocity of the cell adjacent to the boundary.

The term subtracted from ETH for the boundary at the right is

$$\frac{P_{(k)} + P_{(\text{cell to the left})}}{2.} u_{(k)} r_{i-\frac{1}{2}} \pi \Delta t DY_{(j)}$$

and the top is

$$\frac{P_{(k)} + P_{(\text{cell below})}}{2.} v_{(k)} \pi(r_i^2 - r_{i-1}^2)\Delta t(.5)$$

and the bottom, if transmittive, is

$$\frac{P_{(k)} + P_{(\text{cell above})}}{2.} v_{(k)} \pi(r_i^2 - r_{i-1}^2)\Delta t(.5)$$

and is added to ETH. K (in the above equations) refers to the border cell.

The velocity terms in the energy equation for those cells at the transmittive boundaries are, at the right = $u_{(k)} r_{i-\frac{1}{2}}$ and the top = $v_{(k)}$.

Rewriting Eq. (1), the mass transport equation in finite difference form results in

$$\frac{\rho_{(k)}^{n+1} - \rho_{(k)}^{n}}{\Delta t} = \left[\frac{r_{i-1}\,\rho_{i-1}\,u_{i-1}}{r_{i-\frac{1}{2}}\,\Delta r_i} - \frac{r_i\,\rho_i\,u_i}{r_{i-\frac{1}{2}}\,\Delta r_i} + \frac{\rho_{j-1}\,v_{j-1} - \rho_j v_j}{\Delta z_{(j)}}\right] \quad (6)$$

where

$$\Delta z_{(j)} = \frac{V_{(k)}}{A_j^z} = \frac{V_{(k)}}{A_{j-1}^z}$$

where A^z for all $j = \pi(r_{(i)}^2 - r_{(i-1)}^2)$,

and

$$V_{(k)} = \text{volume of cell } k = 2\pi\, r_{i-\frac{1}{2}}\, \Delta r_i\, \Delta z_{(j)} \qquad (7)$$

multiply both sides of Eq. (7) by r_i results in

$$V_{(k)}\, r_i = 2\pi r_i\, \Delta Z_j\, r_{i-\frac{1}{2}}\, \Delta r_i$$

or

$$V_{(k)}\, r_i = A_i^r\, r_{i-\frac{1}{2}}\, \Delta r_i \qquad (8)$$

where A^r = area in the direction perpendicular to the Z axis.

And similarly, multiplying Eq. (7) by r_{i-1} results in

$$V_{(k)}\, r_{i-1} = 2\pi r_{i-1}\, \Delta Z_j\, r_{i-\frac{1}{2}}\, \Delta r_i$$

or (9)

$$V_{(k)}\, r_{i-1} = A_{i-1}^r\, r_{i-\frac{1}{2}}\, \Delta r_i$$

Solving Eqs. (8) and (9) for $r_{i-\frac{1}{2}}\, \Delta r_i$ and substituting their values into Eq. (6) results in

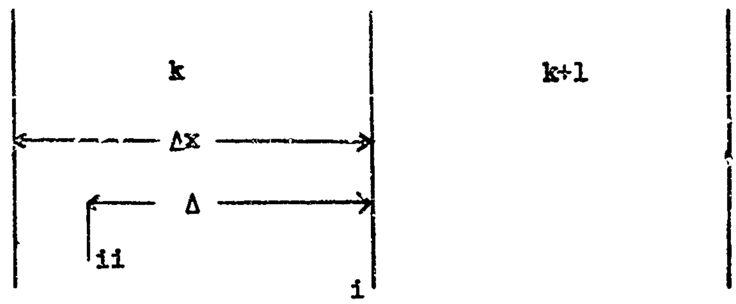
$$\frac{\rho_{(k)}^{n+1} - \rho_{(k)}^{n}}{\Delta t} = \frac{1}{V_{(k)}} \left(A_{j-1}^z\, \rho_{j-1} v_{j-1} - A_j^z\, \rho_j\, v_j + A_{i-1}^r\, \rho_{i-1}\, u_{i-1} - A_i^r\, \rho_i u_i\right)$$

or rewriting in terms of mass,

$$AMX_{(k)}^{n+1} - AMX_{(k)}^{n} = \Delta t\left[(Av)_B^z\, \rho - (Av)_T^z\, \rho + (Au)_L^r\, \rho - (Au)_R^r\, \rho\right]$$

where AMX = mass of cell k in grams and (Av) = area times a velocity (the velocity is a function of the velocity of the two cells in question.) B refers to the bottom, T to the top, L to the left, and R to the right of cell k. The ρ used is the ρ from the cell from which the flux is coming.

Various techniques for velocity weighting in the mass flux have been tried. Results from these trials are presented in Figs. 5 and 6. The scheme presently being used is as follows. Take the r direction as an example:



The mass to move across i is between i and ii where $\Delta = i - ii$; thus $\Delta = \tilde{u}\Delta t$ where $\tilde{u}$ is the weighted velocity at Δ. Using the first two terms of the Taylor series at a distance of $-\Delta$ from i, we expand

$$u_{(i)} = \frac{u_{(k)} + u_{(k+1)}}{2.}$$

or

$$\tilde{u} = \frac{u_{(k)} + u_{(k+1)}}{2.} + (-\Delta)\frac{(u_{(k+1)} - u_{(k)})}{\Delta x}$$

or

$$\frac{\Delta}{\Delta t} = \tilde{u} = \frac{\dfrac{u_{(k)} + u_{(k+1)}}{2.}}{\dfrac{(u_{(k+1)} - u_{(k)})\Delta t}{\Delta x} + 1}$$

if $\tilde{u} > 0$, use $\rho_{(k)}$; if $\tilde{u} < 0$, use $\rho_{(k+1)}$ in the mass flux calculation.

Mass, both components of momentum, and the energy across all four sides of the cell are calculated. By conserving both axial and radial momentum and the total energy, the new velocities are calculated and the new internal energy is then the difference between the total and the kinetic.

A look ahead, two cells in both directions, is done to remove preferential mass transport because of the initial choice of indexing in the r direction first. Take the example where the flux out of the top and right are such that their sum would remove more than the mass in the cell. The code would then assign new fluxes such that the top flux would be its fraction of the total flux out times the mass of the cell, and the right flux would be its fraction of the total flux out times the mass of the cell.

To treat a free surface in the continuous Eulerian scheme, we have chosen to use a density cutoff to limit the mass from flowing through N zones in N time steps. If the mass flow across the free surface results in a density which is less than an input number ($\sim 10^{-3}\ \rho_o$) the flux is held back. To cut the small precursor ahead of the shock front, the velocities are checked against 10^{-8} cm/sh; if they are smaller than this, they are set to zero.

To ensure that the bottom cells in the projectile will empty as the projectile moves up, a scheme using the ρ and v from the cell above is used to calculate the flux. This is continued until the initial velocity of the bottom cell of the projectile begins to change because of the shock. After this point is reached, no special procedure is used for the bottom cells of the projectile.

Boundary Conditions

These cells adjacent to the axis of symmetry ($r = 0$) have the following boundary conditions; the pressure on the left side of the cell is equal to the pressure of the first cell, and the velocity at the left is set to zero. The pressure at the right interface of a cell whose right neighbor is void is zero, and the velocity is that of the occupied cell; similarly for the case of a void cell above.

The pressure and velocity at a transmittive boundary are the following. The pressure at the boundary is set equal to the pressure at the left or bottom boundary respectively for a right and top transmittive boundary (no acceleration of the border cells) while the velocity is set equal to the border cell velocity.

The top and right boundary of the grid is transmittive; the bottom boundary may be either transmittive or reflective, where the same boundary conditions then will exist as for the top or right and the axis of symmetry.

Two passes are used to solve the change in internal energy due to the work terms. The first pass through, one calculates the new velocities at time (n+1) and simultaneously, the velocities at the interfaces at time (n). The time (n) interface velocities are also used to evaluate the internal energy contributions due to terms involving these velocities (see Eq. 4.)

In a second pass the time (n+1) interface velocities are calculated and the associated contributions to the internal energies are computed. (A look-ahead feature of two cells in both directions would enable one to use only one pass.)

An option exists for correcting negative internal energies if they arise in phase 1. The cell where the maximum negative internal energy occurred is recorded; assuming the rate of change of internal energy with time is essentially constant, we calculate a smaller time step, such that the new internal energy will be positive. We complete the entire cycle, to time (n+1), now set the time step negative, integrate backward to time (n), and now forward with the new smaller Δt to a revised time (n+1).

Time Control for Code

The time control for the continuous Eulerian is the same as for the particle in cell scheme, with the exception of the r direction. In the z direction:

$$\Delta m_z = \rho \overline{V} A \Delta t \qquad \text{Assume} \quad \overline{V} = v(k), \quad \rho = \rho(k), \quad \Delta m_z = AMx(k)$$

Then

$$AMx_{(k)} = \frac{AMx(k)}{\pi[r^2_{(i)} - r^2_{(i-1)}]DY(j)} v(k)\pi[r^2_{(i)} - r^2_{(i-1)}]\Delta t$$

$$= AMx(k)v(k)\frac{\Delta t}{DY(j)}$$

or $|v_{(k)}|\Delta t \leq DY(j)$ such that the flux in the z direction will not empty the cell.

In the r direction, the stability is as follows.

$$\Delta m_R = \rho \overline{u} A \Delta t \qquad \text{Assume} \quad \Delta r_R = AMx(k), \quad \overline{u} = u(k), \quad \rho = \rho(k)$$

Then

$$\Delta Mx(k) = \frac{AMx(k)}{2\pi(r_{i-\frac{1}{2}})\Delta r(i)DY(j)} u(k)2\pi r(i)DY(j)\Delta t$$

$$= AMx(k)u(k) \frac{r_{(i)}}{r_{(i-\frac{1}{2})}\Delta r_{(i)}} \Delta t$$

or

$$u(k)\Delta t \leq \frac{\Delta r_{(i)} r_{(i-\frac{1}{2})}}{r_{(i)}} \leq \frac{TAU\ (i)}{2\pi r(i)}$$

C = speed of sound, defined as $\sqrt{\gamma P/\rho}$ for a polytropic form of the equation of state and $(\partial P/\partial\rho)_s^{\frac{1}{2}}$ for a real equation of state. Provisions exist for calling either one.

The time control (Δt) conditions are the Courant condition and that the maximum $[|\frac{u}{\Delta X}|$ and $|\frac{v}{\Delta y}|] < \frac{1}{\Delta t}$.

Three options for time control are:

1. Code will control the Δt, calculated from the Courant and particle velocity scheme, but at a fraction of stability.

2. The Δt loaded at $t = 0$ will remain constant, provided option for integrating backward in time to remove negative internal energies from phase 1 is not operating.

3. Code will control the Δt, decreasing Δt if

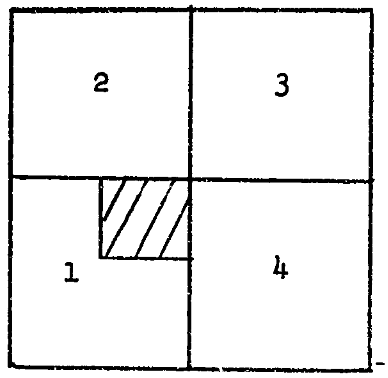
$$\left|\frac{u\Delta t}{\Delta x}\right| \quad \text{or} \quad \left|\frac{v\Delta t}{\Delta y}\right|$$

exceeds FFA (an input number), and increasing Δt if it is less than FFB (an input number.)

The stability check is omitted for a cell if the density of that cell is less than some input number. This prevents isolated debris of high velocity and small masses from controlling the the time step.

Corner Coupling

The question investigated here is the correctness of the mass transport which is done neglecting corner coupling. Below is an example of the comparison with a PIC-like transport:



Assume $u = v$ for all four cells and

$$v\Delta t = \frac{\Delta x \text{ or } \Delta y}{2}$$

Where particles in the PIC scheme located in the shaded area will cross into zone 3 in one time step, the OIL code requires two time steps for mass to move into zone 3, first by the path of zone 1 to 2 or 4, and finally to zone 3.

In the case of very small time steps, it is seen that the above approximation is unimportant. We have chosen to run most of our problems at .5 stability. Further, from early test runs of an impact calculation, results did not change appreciably as the factor was varied from one-fourth to one-half to nine-tenths of stability.

The PIC Transport

The changes required to change from a continuous mass transport to the particle transport are:

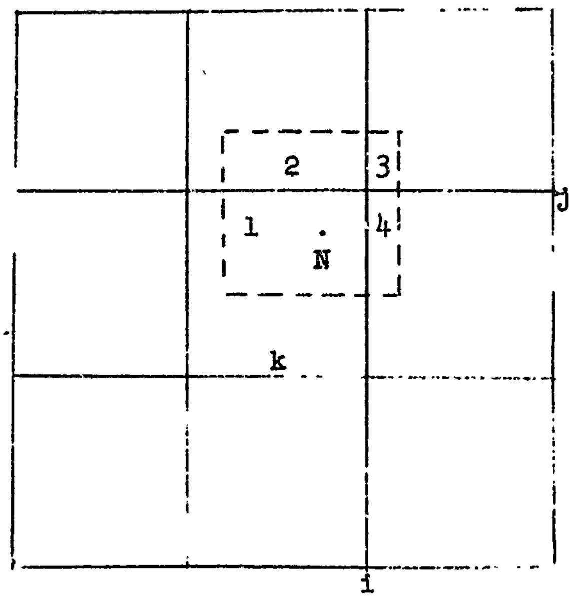
1. The problem number must be negative.
2. The transport and rezone subroutines must be replaced with those for PIC transport.

Following is a brief discussion of the particle (PIC) transport characterizing SHELL. Two scratch tapes are required for the particles. SHELL reads in particle records from one tape, processes the particles updating the coordinates, writes them out on the other tape, and interchanges tape numbers.

Five variables are associated with each particle; the mass (AM), the r coordinate (XL), and Z coordinate (YL), the i-coordinate of the cell where the particle lies (iwl), and the j-coordinate of the cell where the particle

lies (iw2). Thus, one computes the cell number where the particle is by k = (j-1)*IMAX + i + 1.

The particles are moved with an area weighted velocity, which is basically a cell placed with the particle at the center. The overlay of this cell on the four cells in question times the velocity of that cell is summed for the four cells and then the weighted velocity is calculated by dividing through by the total area.



Example: For particle N the u component of velocity used to move the particle is

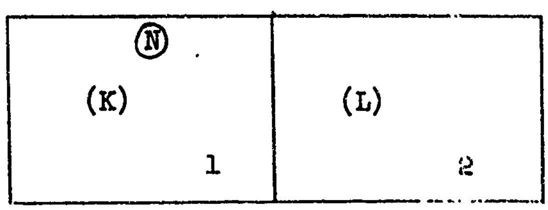
$$\bar{u} = \frac{\sum_{i=1}^{4} A_i u_i}{\sum_{i=1}^{4} A_i}$$

and the v component is

$$\bar{v} = \frac{\sum_{i=1}^{4} A_i v_i}{\sum_{i=1}^{4} A_i}$$

The particle is then moved with the area weighted velocities. If the particles does not leave the cell (k), no additional calculations are needed; process the next particle. If the particle leaves cell (k), it carries with it a mass, momentum, and internal energy.

By conserving momentum and total energy, one then changes the quantities in the new cell. No changes except removing the particle mass from the mass of cell (k) are necessary for updating cell (k).



Example: Particle N moves from cell (k) to cell (L). Conserving both axial and radial momentum:

$$(\bar{M}_2)\bar{u}_2 = M_2u_2 + mu_{11}$$

and

$$(\bar{M}_2)\bar{v}_2 = M_2v_2 + mv_{11} ,$$

where the line above signifies the updated variable. Thus,

$$\bar{u}_2 = \frac{M_2u_2}{\bar{M}_2} + \frac{mu_{11}}{\bar{M}_2}$$

$$\bar{v}_2 = \frac{M_2v_2}{\bar{M}_2} + \frac{mv_{11}}{\bar{M}_2}$$

However, M_2 is not available at this stage (since $\bar{M}_2$ has replaced M_2); substitute $M_2 = \bar{M}_2 - m$ results in $\bar{u}_2 = \frac{m}{\bar{M}_2}(u_{11} - u_2) + u_2$; similarly for the v component. Note that u_{11} and v_{11} are set equal to u_1 and v_1 unless there has been an elastic bounce off a reflective boundary (requiring the velocities to change sign to conserve momentum); then u_{11} and v_{11} are set to $-u_1$ and $-v_1$.

To calculate the new internal energy in cell (2) requires that we conserve total energy and momentum, resulting in the expression:

$$M_1I_1 + \tfrac{1}{2}M_1(u_1^2 + v_1^2) + M_2I_2 + \tfrac{1}{2}M_2(u_2^2 + v_2^2) =$$

$$(M_1 - m)I_1 + \tfrac{1}{2}(M_1 - m)(u_1^2 + v_1^2) + (M_2 + m)\bar{I}_2 + \tfrac{1}{2}(M_2 + m)(\bar{u}_2^2+\bar{v}_2^2)$$

Solving for $\bar{I}_2\bar{M}_2$ =

$$mI_1 + \tfrac{1}{2}m(u_1^2 + v_1^2) + M_2I_2 + \tfrac{1}{2}M_2(u_2^2 + v_2^2) - \tfrac{1}{2}\bar{M}_2(\bar{u}_2^2 + \bar{v}_2^2)$$

Substituting the new values for $\bar{u}_2$ and $\bar{v}_2$ from conserving momentum results in

$$\bar{I}_2 = I_2 + \frac{m}{\bar{M}_2}\left\{I_1 - I_2 + \left[\frac{(u_{11} - u_2)^2 + (v_{11} - v_2)^2}{2.}\right]\left(1 - \frac{m}{\bar{M}_2}\right)\right\}$$

Thus, after each particle is moved, the two cells involved are updated. After a particle record has been processed, the particles, with their new coordinates and i and j value of the new cell location, are then written on another tape, which will become the starting conditions for the next transport cycle. An option exists to call rezone if particles leave the top or the right of the grid.

Viscosity

The movement of mass across the cell boundaries give rise to force which is effective in reducing fluctuations that arise from the differencing technique(ref 1). This is of the form of a "true" viscosity, being proportional to the velocity gradient. This viscosity is present at all times, both in compressions and rarefactions. No additional (that is, a controllable) artificial viscosity is present in this version of oil.

3.2. Special Subroutines

The REZONE routine for the particle transport is different from that required for the continuous mass transport version. If material leaves the grid out of the right or top, a trigger is set to call REZONE.

The REZONE multiplies all dimensions by two, so that four old cells become one in the new grid. The total number of cells remains the same, and the target is doubled in depth and width by adding mass at the sides and back surface. This scheme does conserve total energy, and by conserving momentum, also, new velocities and internal energy are calculated. This will cause the total internal energy of the system to rise slightly.

For the particle rezone, it is not necessary to multiply all dimensions by two, but rather change the Δx's and Δy's by any prescribed amount. One adds new material with the same density, internal energy, and velocity distributions as are available in CLAM. The number of particles per cell to add is also an input number. For a more complete description, see Reference 3.

A subroutine SETUP is available to generate the initial grid (bypasses the generator code CLAM) if both the target and projectile are of the same density. The projectile must be a right circular cylinder. This routine assumes that all Δx's are the same and all Δy's are the same. An asterisk before the symbol implies it is floating point.

SYMBOL	LOCATION	DESCRIPTION
* Z(111)	111	Initial density g/cm^3
* Z(112)	112	Initial pellet velocity cm/sh.
* DX(1)	7845	Δx in cm
* DY(1)	7897	Δy in cm
iMAX	33	Maximum number of zones in the r direction
jMAX	35	Maximum number of zones in the z direction
i1	47	The i value of the radius of the projectile +2
i2	48	The j value of the top of projectile +2
* PROB	1	Any positive number for the problem
* PK(3)	237	Must be a positive number

* PK(4)	238	Set = 1
* PK(5)	239	Right boundary (i) of projectile
* PK(6)	240	Bottom (j)+1 of projectile
* PK(7)	241	Top (j) of projectile
* PK(8)	242	Set = 1
* PK(9)	243	Right (i) boundary of target
* PK(10)	244	Bottom (j)+1 of target
* PK(11)	245	Top (j) of target

And the usual input data to start OIL code from a CLAM tape.

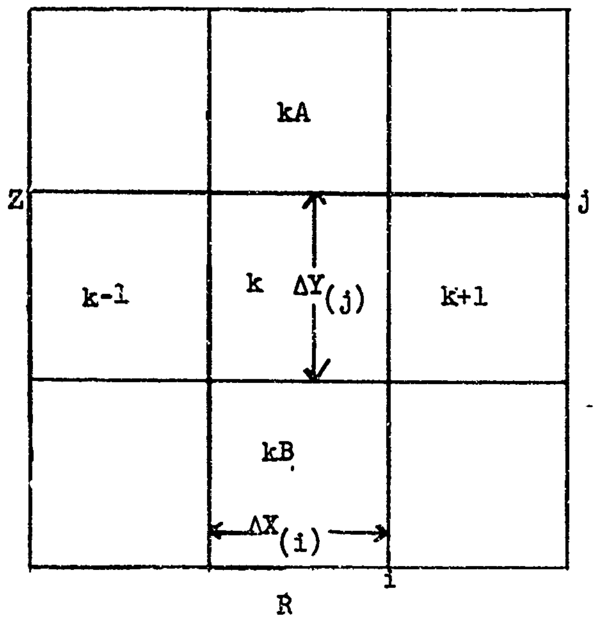
An Example of Input for OIL Using the Subroutine SETUP.

```
(OIL   INPUT)
   GAS   GAMMA= 1.5 RHO TARGET = RHO PELLET =1.
1   23531.         0.        0.
    11211.01
      111.
    11111.
     7511.     -3
    13821.     -2.5
    23871.         6.        9.        20.       1.        24.       21.
    245132.
2    33124.
2    35132.
   784511.
   789711.
     7711.
     271-1.
     821-1.
     651-1.
     7111.
     1421.         .5
     2421.     -50.
     8621.5        1.5
    147120.
2    4728.         22.
    11311.     -2
1     3510.        4.        4.        16.       16.
1    7211.     +4
```

This will make a binary tape that is equivalent to that made by the example in CLAM (see Section 2.1). See Section 5.2 for format for CARDS subroutine.

3.3. Logic of OIL

The logic involved in following a given cell (K) from time t to t+Δt, or from cycle n to n+1.



$$k = (j-1)iMAX + i + 1$$

$$TAU(i) = \pi(x^2_{(i)} - x^2_{(i-1)})$$

Fig. 3

1. CDT Routine

Here one calculates the pressure (P) for cell (k) where $P(k) = f[\rho_{(k)}, I_{(k)}]$ where $\rho_{(k)} = \frac{AMZ(k)}{TAU(i)DY(j)}$. The speed of sound $(\gamma P/\rho)^{\frac{1}{2}}$ or $(\partial P/\partial \rho)_s^{\frac{1}{2}}$ is then calculated and the Courant condition for stability, that $\frac{C\Delta t}{\min\,(\Delta x \text{ or } \Delta y)} < \frac{1}{2(\gamma_{max-1})}$ and the particle velocity criteria that the $\max\,[\left|\frac{u}{\Delta x}\right| \text{ and } \left|\frac{v}{\Delta y}\right|] < \frac{1}{\Delta t}$ are calculated. From these stability checks a new Δt is calculated or Δt remains the same (see options under description of common for OIL, symbol CABIN.) The cycle number and the time are now advanced. (A radiation time step is also calculated, although radiation is not being used in this version of the code.)

2. EDIT Routine

The OIL code has four separate editing-like routines all included in the routine called EDIT. A section called short print displays the time, cycle number, the total internal and kinetic energy and total mass, and various other integral quantities such as momenta , and mass in various

angles (see Sect. 3.4.) A plot routine is also available in the EDIT routine which places an (x) (in an equal cell size grid, corresponding to the actual grid in OIL) if there is any mass in the cell; thus it is a display of mass movement from time to time. A long print routine may also be called for that edits on each page a column from the OIL grid which contains the coordinates of the column, and the cell quantities as a function of the row coordinates (see column identification in Sect. 5.2 FORTRAN listing of EDIT.)

The last option is a dump routine which dumps all necessary data for restarting the problem. This data may also be used for the automatic plotting routines.

Various input numbers (see Sect. 3.4) specify the frequency that these routines will be called for, and an input number specifying a cycle number to stop.

In the following discussion, please refer to Fig. 3.

3. PH1 Routine

Here we integrate the two momentum equations and the change in internal energy due to the work terms. No material is moved at this time, and the transport terms are dropped. Using the new pressures and the time step which were computed in CDT, we now prepare to integrate the equations.

PL(j), the pressure at interface (i-1) and uL(j), the velocity at interface (i-1) are available from the previous column sweep on i-1.

$$PL(j) = \frac{P(k) + P(k-1)}{2.}$$

$$uL(j) = \frac{r_{i-3/2}\, u^n_{(k-1)} + r_{i-\frac{1}{2}}\, u^n_{(k)}}{2.}$$

The PBLO term, which was the PABOVE for cell (kB) and VBLO, VABOVE for cell kB, is also available for interface j-1.

$$PBLO = \frac{P(k) + P(kB)}{2.}$$

$$VBLO = \frac{v(k) + v(kB)}{2.}$$

Now we calculate terms at interface i and j:

$$PR = \frac{P(k+1) + P(k)}{2.} \qquad URR = \frac{r_{i-\frac{1}{2}}\, u(k)^n + r_{i+\frac{1}{2}}\, u(k+1)^n}{2.}$$

$$P_{ABOVE} = \frac{P_{(k)} + P_{(kA)}}{2.} \qquad VABOVE = \frac{v(k)^n + v(kA)^n}{2.}$$

Now we can integrate the two momentum equations

$$\rho \frac{\partial u}{\partial t} = -\frac{\partial P}{\partial r}$$

or

$$u^{n+1}_{(k)} = u^n_{(k)} + \left(\frac{PL_{(j)} - PRR^n}{AMX(k)}\right) 2r_{i-\frac{1}{2}}\, \pi \Delta t DY(j)$$

and

$$\rho \frac{\partial v}{\partial t} = -\frac{\partial P}{\partial z}$$

or

$$v^{n+1}_{(k)} = v^n_{(k)} + \left(\frac{PBLO - PABOVE}{AMX(k)}\right) \pi(r^2_i - r^2_{i-1})\Delta t$$

Now one can add the work term due to velocities at cycle N to the change in internal energy.

$$I^{n+\frac{1}{2}}_{(k)} = I^n_{(k)} + \frac{P(k)}{AMX(k)} \Big[(uL^n_{(j)} - uRR^n)\, \pi \Delta t DY(j)$$

$$+ \left(\frac{VBLO^n - VABOVE^n}{2.}\right) \pi(r^2_i - r^2_{i-1})\Delta t\Big]$$

$$\rho \frac{\partial I}{\partial t} = -P\left(\frac{\partial v}{\partial z} + \frac{1}{r}\frac{\partial ur}{\partial r}\right)$$

Now we make one more pass through the grid, this time omitting the momentum equations but calculating the velocity terms at the interface, where again we only have to calculate the data at interface i and j:

$$uL_{(j)} = \frac{r_{i-3/2}\, u^{n+1}_{(k-1)} + r_{i-\frac{1}{2}}\, u^{n+1}_{(k)}}{2.} \qquad VABOVE = \frac{v^{n+1}_{(k)} + v^{n+1}_{(kA)}}{2.}$$

$$uRR = \frac{r_{i-\frac{1}{2}}\, u^{n+1}_{(k)} + r_{i+\frac{1}{2}}\, u^{n+1}_{(k+1)}}{2.} \qquad VBLO = \frac{v^{n+1}_{(k)} + v^{n+1}_{(kB)}}{2.}$$

and then add to $I^{n+\frac{1}{2}}_{(k)}$ the work terms due to velocities at cycle n+1.

$$I^{n+1}_{(k)} = I^{n+\frac{1}{2}}_{(k)} + \frac{P_{(k)}{}^{n}}{AMX(k)} \Big[(uL^{n+1}_{(j)} - uRR^{n+1})\, \pi \Delta t DY(j)$$

$$+ \Big(\frac{VBLO^{n+1} - VABOVE^{n+1}}{2.}\Big)\, \pi(r_i^2 - r_{i-1}^2)\Delta t\Big]$$

The specific internal energy is checked during both passes for negative values. If a negative value is found, we assume dI/dt is constant over the time step, and recompute a new Δt (not placing it in the Δt storage) that will prevent I from going negative. After the completion of integrating all values to cycle n+1, an option exists for removing these negative energies. If one selects the option, Δt is set equal to -Δt, the code integrates backward two passes to cycle n, then replacing Δt with the new, smaller Δt and now forward in time with two passes using the smaller Δt.

4. <u>PH2 Routine</u>

Here we move the mass and approximate the transport terms in the momentum and energy equations that were omitted in Phase 1.

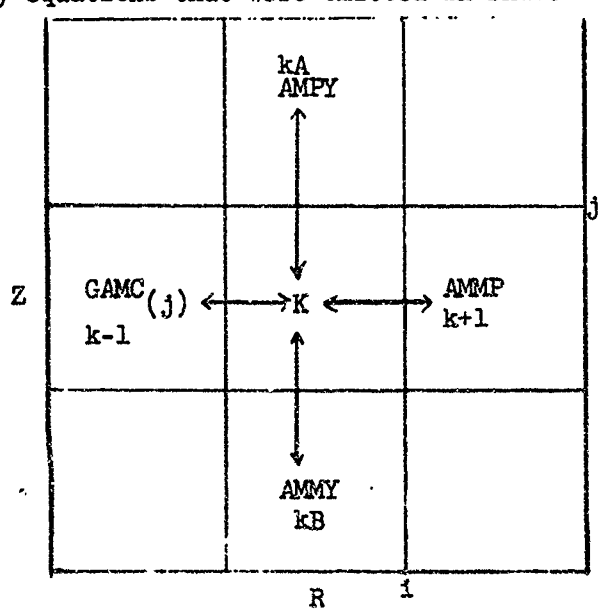


Fig. 4

AMMP = mass flow across the right boundary
AMUR = radial momenta across the right boundary
AMVR = axial momenta across the right boundary
DELER = total specific energy across the right boundary

AMMY = mass flow across the bottom boundary
AMMU = radial momenta across the bottom boundary
AMMV = axial momenta across the bottom boundary
DELEB = total specific energy across the bottom boundary
GAMC(j) = mass flow across the left boundary
FLEFT(j) radial momenta across the left boundary
YAMC(j) = axial momenta across the left boundary
SIGC(j) = total specific energy across the left boundary
AMPY = mass flow across the top boundary
AMUT = radial momenta across the top boundary
AMVT = axial momenta across the top boundary
DELET = total specific energy across the top boundary

Again, following a typical cell k (Fig. 4) the masses, the momentas, and the energies are now available at the left and bottom boundaries of cell (k) from the previous column sweep and the cell below.

Now, begin by calculating the mass flow at the top of cell (k).

$$V_{ABOVE} = \frac{V_{(k)} + V_{(kA)}}{2.}$$

then form

$$\frac{V_{ABOVE}}{1 + \left[\frac{V_{(kA)} - V_{(k)}}{\Delta y(j)}\right] \Delta t}$$

as the weighted velocity to use in the flux equation and store it in V_{ABOVE}. If it is positive, use $\rho_{(k)}$; if it is negative, we use $\rho_{(KA)}$. Now calculate the mass flow across the top as $\Delta m_T = \rho_{(M)} A_{(j)} \bar{V} \Delta t$ where

$$\bar{V} = V_{ABOVE}$$
$$M = \text{donor cell}$$

or

$$\Delta mT = AMPY = \frac{AMx(m)}{DY(j)} V_{ABOVE}(\Delta t)$$

Now we calculate the mass flow at the right boundary of cell k. uRR is defined as $\frac{u(k) + u(k+1)}{2.}$ then form

$$\frac{URR}{1 + \left[\frac{u_{(k+1)} - u(k)}{\Delta x(i)}\right]\Delta t}$$

as the weighted velocity to use in the flux equation and store it in uRR. The mass flow across i is then

$$\Delta m_R = \rho_{(M)}A_{(i)}\bar{u}\,\Delta t$$

where

$$\bar{u} = uRR$$

$$N = \text{i value of donor cell}$$

$$M = \text{donor cell}$$

$$PIDTS = \frac{1}{\pi\Delta t}$$

or

$$\Delta m_R = AMMP = \frac{AMx(M)}{TAU(N)} \frac{x(i)}{PIDTS} 2(uRR)$$

Now check to see if these masses will more than empty the cell, since it is possible that the left and bottom flux were both negative. Search cells ahead in both directions to remove preferential mass movement. As an example, suppose the flux at the TOP (AMPY) is positive and the flux at the right is positive, where their sum is larger than the mass in cell (k); then normalize the fluxes the following way. The flux out of the top is

$$\frac{F_T[AMX(k)]}{F_T + F_R}$$

and the flux out of the right is equal to

$$\frac{F_R[AMX(k)]}{F_T + F_R}$$

where F is a symbol for the mass flux.

The momenta associated with these masses are now computed. The sign of the flux specifies the zone where the mass came from, thus at the top; the radial component equals AMUT = AMPY(u(N)) and the axial AMVT = AMPY(v(N)) where (N) = cell number of the donor cell.

The momenta for the right is AMUR = AMMP(u(N)) for the radial and AMVR = AMMP(v(N)) for the axial component, where again (N) = cell number of the donor cell.

The total specific energy that those mass fluxes carry is also calculated at this time; for the top it is equal to

$$\text{DELET} = I_{(n)} + \frac{u^2_{(N)} + v^2_{(N)}}{2.}$$

and for the right it is equal to

$$\text{DELER} = I_{(n)} + \frac{u^2_{(N)} + v^2_{(N)}}{2.}$$

where again (N) = cell number of the donor cell.

The mass now in cell (k) is equal to DELM = AMX(k) + GAMC(j) + AMMY - AMPY - AMMP which equals the original mass plus the mass flow across the left, the bottom, and less the mass flow across the top and the right.

The total axial momenta that have come into or left cell (k) is = SIGMV = YAMC(j) + AMMV - AMVT - AMVR = the momenta crossing the left boundary plus the momenta crossing the bottom boundary less the momenta crossing the top and the right boundary.

The total radial momenta that has come into or left cell (k) is SIGMU = FLEFT(j) + AMMU - AMUT - AMUR = momenta crossing the left and bottom boundary less the momenta crossing the top and the right.

Similarly we calculate the total energy that these fluxes have carried = DELEK = $(\text{GAMC}_{(j)})\ \text{SIGC}_{(j)}$ + (AMMY) Deleb - (AMPY) Delet - (AMMP) Deler = the mass times the total specific energy at the left plus the similar term for the bottom loss the similar terms for the top and the right.

Now by conserving momenta and total energy, calculate the new specific internal energy and velocities of cell (k).

$$MU_L + MU_B - MU_T - MU_R + MU_k = (\text{TOTAL MASS})\bar{u}$$

where total mass = DELM and

$$Mv_L + Mv_B - Mv_T - Mv_R + Mv_k = (\text{TOTAL MASS})\bar{v}$$

and the new specific internal energy

$$I_{(k)} = \frac{E_L + E_B - E_T - E_R + E_k}{\text{DELM}} - \frac{\bar{u}^2 + \bar{v}^2}{2.}$$

and now AMX(k) set = to DELM.

The subscripts L, B, T, R, refer to the left, bottom, top, and right. Now the values that we calculated at the right for cell (k) are now set to the left values for cell (k+1) and the top values for cell (k) now become the bottom values for cell (kA).

The limits of the DO loops on i and j are i1 and i2. A check is done to see if mass or energy has moved beyond i1 or i2 and then the counter is increased by 1. This check is also done in PH1. By using this scheme, we pr ss only the active mesh at all times, in all the subroutines.

3.4. List of Common (OIL)

The location refers to the location of that symbol relative to the beginning of common. Since the beginning of common is assigned the same location for each subroutine, a program (CARDS) is available for changing any word in common. The z block is first in common for OIL.

Note that if one should change the dimensions of the arrays, he must be careful and also make the necessary changes of the locations in the following tables.

Symbol	Location	No. of Words	Units	Description
AID	706	1	--	Not used, since this is a one-material (x) code
AIX	707	3500	jerks/g	Specific internal energy (x) for cell (k)
AM	4207	130	g	Mass of particle (N) for PIC transport only
AMD	4337	1	--	Not used in this one-material code
AMK UR(16)	220	15	g	Storage (EDIT) for summing masses in given angles for editing
AMX	4338	3500	g	Total (x) mass in cell k
AREA	7838	1		Tag, used in PH2 (EUL and PIC)
BIG	7839	1	--	Not used
BOUNCE	7840	1	--	Tag used in particle PH2
CABLN	82	1	--	If < 0 code controls Δt but at Z(139) of instability
	Caution: You must load a Δt for this option			If = 0, code will control the Δt, decreasing Δt if $\left\lvert\frac{u\Delta t}{\Delta x}\right\rvert$ or $\left\lvert\frac{v\Delta t}{\Delta y}\right\rvert$ exceed FFA (an input number) and increasing Δt if $\left\lvert\frac{u\Delta t}{\Delta x}\right\rvert$ or $\left\lvert\frac{v\Delta t}{\Delta y}\right\rvert$ is less than FFB (an input number)
	This holds provided SN $\neq$ 0			If > 0, DT loaded will remain constant.
DDXN	7841	1	--	Not used
DDVK	7842	1	--	Not used
DKE	7843	1	--	Not used
DVK	7844	1	--	Not used
DX	7845	52	cm	DX(i) = X(i) - X(i-1)
DY	7897	100	cm	DY(j) = Y(j) - Y(j-1)
E	7997	1	--	Not used
FD	7998	1	--	Not used
FS	7999	1	--	Flag in PH2
FX	8000	1	--	Not used
OUT	8001	1	--	Tag in particle PH2
P	8002	3500	jerks/cm^3	Material pressure in cell k
PABOVE	11502	1	jerks/cm^3	$\frac{P_{(k)} + P_{(\text{cell above})}}{2.}$

PBLO	11503	1	jerks/cm^3	$\frac{P_{(k)} + P_{(\text{cell below})}}{2.}$
PiDTS	11504	1	1/cm.sh	$= \frac{1.}{\Delta t\ \pi\ DY(j)}$ in (PH1); $\frac{1}{\pi \Delta t}$ in PH2
PK UR(31)	235	15	g.cm/sh	Radial momentum in certain angles for editing
PL, PR, GAMC PR(100), SIG(C)	405	200	Many	Pressure in PH1, flux in PH2, etc.
PPABOV	11505	1	--	Not used
PRR	11506	1	jerks/cm^3	$\frac{P_{(k)} + P_{(\text{cell to the right})}}{2.}$
PUL	11507	1	--	Not used
QDT	11508	1	--	Not used
QK	250	15	g.cm/sh	Axial momentum in certain angles for editing
RC	11509	1	cm	$[(X(i) + X(i-1)]/2.$ in PH1
ReZ	11510	1	--	If > 0 and ReZFcT > 0, then PH2 (EUL) will call subroutine REZONE, ReZ set in PH2.
RHO	11511	1	g/cm^3	Density
RL	11512	1	--	Not used
RR	11513	1	cm	$[X(i) + X(i+1)]/2$ in PH1
SIG	11514	1	cm	Minimum Δx or Δy in CDT routine
SIGN = QOOOFL	11515	1	0	Not used
SWITCH	11516	1	--	Not used
TAB	205	15	--	Tan(α) (Table of, used in EDIT routine)
TABLM	11517	1	--	Not used
TAU	11518	52	cm^2	$= \pi(X^2_{(i)} - X^2_{(i-1)}) =$ area
TAUDTS	11570	1	cm^{2}sh	$=$ TAU(i)Δt in PH1.
TAUDTX	11571	1	--	Not used
U	11572	3500	cm/sh	R component of velocity in cell (k)
UK	15072	1	cm/sh	R component of velocity in cell (k) used in PIC transport

UL, UR, Fleft etc.	205	200	cm^2/sh	$\frac{U(k)RC + U(k-1)RCC}{2.}$ where $RCC = \frac{X_{(i-1)} + X_{(i-2)}}{2.}$ and $RC = \frac{X_{(i)} + X_{(i-1)}}{2.}$
URR	15073	1	cm^2/sh	[U(k)RC + U(k+1) RR]/2.
UT	15074	1	--	Signal in PH1 (decrease Δt)
UU	15075	1	sh	New Δt if PH1 integrates back for I < 0, set originally to 10^{15}.
UUU	15076	1	--	Not used
UTEF	15077	1	cm/sh	R velocity component used to move particle when using PIC transport
UVMAX	15078	1	1/sh	|Max velocity|/Min(ΔX or ΔY)
V	15079	3500	cm/sh	Axial (Z) component of velocity for cell (k)
VABOVE	18579	1	cm/sh	[V(k) + V(cell above)]/2.
VBLO	18580	1	cm/sh	[V(k) + V(cell below)]/2.
VEL	18581	1	--	Used as a tag in PH1 and Max(γ-1) in CDT, and tag in EUL. PH2
VK	18582	1	cm/sh	Axial component of velocity in cell (k) for PIC transport
VT	18583	1	--	Not used
VTEF	18584	1	cm/sh	Z velocity component used to move particle when using PIC transport
VV	18585	1	--	Not used
VVABOV	18586	1	--	Not used
VVBLO	18587	1	--	Not used
W2	18588	1	--	Not used
W3	18589	1	--	Not used
WPS	18590	1	--	Working Storage
WS	18591	1		
WSA	18592	1		
WSB	18593	1		
WSC	18594	1		

XX	152	53	cm	XX(2) = X(1)
XL	18595	130	cm	R coordinate of particle N
XLF	18725	1	--	Fraction of area on left to use in velocity weighting for PIC PH2.
XN	18726	1	cm	R coordinate of particle N at cycle (n-1)
XR	18727	1	--	Fraction of area to the right to use in velocity weighting for PIC PH2.
X1	78	1	--	Not used
X2	79	1	--	Not used
Y	606	100	cm	Y(j) = top dimension of zone (i,j)
YL	18728	130	cm	Z coordinate of particle N
YLW	18858	1	--	Fraction of area below to be used in velocity weighting for PIC PH2.
YN	18859	1	cm	Z coordinate of particle N at cycle (n-1)
YU	18860	1	--	Fraction of area above to be used in velocity weighting for PIC PH2.
YY (YY(2) = Y(1)	605	1	cm	YY(j+1) = Y(j)
Y1	80	1	--	Not used
Y2	81	1	--	Not used
Z	1	150	--	See pages where Z(1) through Z(150) are defined
ZMAX	18861	1	cm	An up-to-date value of largest Y coordinate of all particles used in PIC PH2.
i	18862	1	--	Index (Working Storage)
ii	18863	1		
iN	18864	1		
iR	18865	1		
iWS	18866	1		
iWSA	18867	1		
iWSB	18868	1		
[illegible]C	18869	1	--	
i1	47	1	--	The right boundary of the active grid + 2, MAX(i1) = iMAX

i2	48	1	--	The top boundary of active grid + 2 MAX(i2) = jMAX
i3	49	1	--	Not used
i4	50	1	--	Not used
iW1	18870	130	--	= (i) of the cell (k) where particle (N) is located, used in PIC PH2.
j	19000	1	--	Index (Working Storage)
jN	19001	1		
jP	19002	1		
jR	19003	1		
K	19004	1	--	Index of cell defined such that k = (j-1) iMAX + i+1
KDT	41	1	--	If KDT = 0, Δt has changed if ≠ 0 Δt remains constant
KN	19005	1	--	Index (Working Storage)
KP	19006	1		
KR	19007	1		
KRM	19008	1		
L	19009	1		
M	19010	1		
MA	19011	1		
MB	19012	1		
MC	19013	1		
MD	19014	1		
ME	19015	1		
MZ	19016	1	--	Set by input, used in EDIT for length of Z block to write on tape, present value = 150
N	19017	1	--	Index(Working Storage)
NK	19018	1		
NKMAX	19019	1	--	
NK1	19020	1		
NO	19021	1		
NR	19022	1	--	Maximum number of radiation cycles/ hydro cycle calculated in CDT NR ≦ NRM

iW2	19023	130	--	= (j) value of cell (k) where particle (N) is located, used in PIC PH2.
iZ	1	150	Many	Fixed value of Z block
Fleft	205	100		Equivalenced, used for flux terms
YAMC	304	100		in Eulerian continuous (PH2)
SIGC	504	100		

- - - - - - - - - - - - - - - -

Location	Symbol	Units	Description
Z(1)	PROB	--	Problem number (if positive, this is an OIL run; if negative, this is a PIC run
Z(2)	CYCLE	--	Cycle number (floating point value)
Z(3)	DT	sh	Δt hydro = $t^n - t^{n-1}$
Z(4)	PRINTS	--	Cycle frequency for short print
Z(5)	PRINTL	--	Cycle frequency for long print
Z(6)	DUMPT7	--	Cycle frequency for binary tape dumps
Z(7)	CSTOP	--	Cycle number at which problem stops
Z(8)	PIDY	--	= π = 3.1415927
Z(9)	TMZ	g	Total (X + ·) mass at t = 0 (calculated in CLAM
Z(10)	GAM	--	Not used
Z(11)	GAMD	--	$1./(\gamma_{\cdot}-1)$ Computed in Input
Z(12)	GAMX	--	$1./(\gamma_x-1)$
Z(13)	ETH	jerks	Total energy (computed in CLAM for t=0.) Changed in PH1 at transmittive boundaries and in PH2 if mass leaves the system, and by radiation flow out of the system.
Z(14)	FFA	--	Upper limit for stability and to calculate Δt, only if CABLN = 0.
Z(15)	FFB	--	Lower limit for stability and to calculate Δt, only if CABLN = 0.
Z(16)	TMDZ	g	Total (·) mass (t = 0) calculated in CLAM
Z(17)	TMXZ	g	Total (x) mass (t = 0) calculated in CLAM
Z(18)	XMAX	cm	= X (iMAX)
Z(19)	TXMAX	cm	2 (XMAX) t = 0. calculated in CLAM
Z(20)	TYMAX	cm	2 (YMAX) t = 0. calculated in CLAM
Z(21)	AMDM	g	Min(·) particle mass/2.; calculated in CLAM

Z(22)	AMXM	g	Min (x) particle mass/2. Calculated in CLAM
Z(23)	DNN	--	$(ETH - E)^{n-NPC}/ETH$
Z(24)	DMIN	--	IF (ECK) Note Z(76) > DMIN, problem will stop and the edit routine will call dump.
Z(25)	FEF	--	Not used
Z(26)	DTNA	sh	Δt^{n-1}
Z(27)	CVIS	--	If < 0, bottom boundary is transmittive; otherwise reflective boundary.
Z(28)	NPR	--	Index (Working storage)
Z(29)	NPR1	--	" "
Z(30)	NC	--	Cycle number in fixed point.
Z(31)	NPC	--	Number of cycles between short prints
Z(32)	NRC	--	Index
Z(33)	iMAX	--	Maximum number of zones in R direction
Z(34)	iMAXA	--	iMAX + 1
Z(35)	JMAX	--	Maximum number of zones in Z direction
Z(36)	JMAXA	--	jMAX + 1
Z(37)	KMAX	--	(iMAX)(jMAX) + 1
Z(38)	KMAXA	--	KMAX + 1
Z(39)	NMAX	--	Total number of particles + 1, generated in CLAM for PIC problem only.
Z(40)	ND	--	Total number of (·) particles + 1 generated in CLAM
Z(41)	KDT	--	Defined previously
Z(42)	ixMAX	--	Not used
Z(43)	NOD	--	Index
Z(44)	NOPR	--	Index
Z(45)	NiMAX	--	New iMAX before adding new zones
Z(46)	NjMAX	--	New jMAX before adding new zones
Z(47)	i1	--	Defined previously
Z(48)	i2	--	Defined previously
Z(49)	i3	--	Not used
Z(50)	i4	--	Not used
Z(51)	N1	--	Scratch tape number for particles if this is a PIC run.
Z(52)	N2	--	Scratch tape number for particles if this is a PIC run.

Z(53)	N3	--	Number of particle records generated if this is a PIC run.
Z(54)	N4	--	Number of particles-1 per record (MAX = 127) if this is a PIC run.
Z(55)	N5	--	Not used
Z(56)	N6	--	Number of particles on last particle record if this is a PIC run
Z(57)	N7	--	Not used
Z(58)	N8	--	Not used
Z(59)	N9	--	Not used
Z(60)	N10	--	= i value of zone that is controlling Δt
Z(61)	N11	--	= j value of zone that is controlling Δt
Z(62)	NRM	--	= maximum number of Rad cycles/Hydro (input number)
Z(63)	TRAD	sh	$NR \cdot \Delta t$ Rad = ΔT Hydro; not used in this version
Z(64)	XNRG	jerks	Total energy of (x) material
Z(65)	SN	--	If = 0 code will decrease Δt to correct for $I < 0$, if $\neq 0$, those $I < 0$ are left alone
Z(66)	DXN	--	Not used
Z(67)	RADER	g-cm/sh	Total positive radial momentum (x only)
Z(68)	RADET	g-cm/sh	Total positive axial momentum (x only)
Z(69)	RADEB	g-cm/sh	Total positive radial momentum (x) for material under target
Z(70)	DTRAD	--	Not used
Z(71)	REZFCT	--	If = 0, PH2 will not trigger rezone
Z(72)	RSTOP	--	Not used in continuous version
Z(73)	SHELL	--	Not used
Z(74)	BBOUND	--	Not used in this version
Z(75)	TOZONE	g/cm^3	Minimum density for mass flow at free surface
Z(76)	ECK	energy check	$\left[\left(\frac{ETH - E}{ETH}\right)^{n} - \left(\frac{ETH - E}{ETH}\right)^{n-NPC}\right] / NPC$
Z(77)	SBOUND	--	Fraction of Δ in mass weighting velocity EUL PH2 ~ 1.0
Z(78)	X1	--	Not used
Z(79)	X2	--	Not used
Z(80)	Y1	--	Not used

Z(81)	Y2	--	Not used
Z(82)	CABLN	--	Already defined
Z(83)	VISC	--	Not used
Z(84)	T	sh	Total time up to cycle N, $t^n = t^{n-1} + \Delta t$
Z(85)	GMAX	--	Maximum of γ_x or γ.
Z(86)	WSGD	--	γ.
Z(87)	WSGX	--	γ_x and $(\gamma_{max} - 1)$ in CDT
Z(88)	GMADR	--	$\gamma./(\gamma. - 1)$
Z(89)	GMAXR	--	$\gamma_x/(\gamma_x - 1)$
Z(90)	S1	--	Not used
Z(91)	S2	--	Not used
Z(92)	S3	--	Not used
Z(93)	S4	--	Not used
Z(94)	S5	--	Not used
Z(95)	S6	--	Not used
Z(96)	S7	--	Not used
Z(97)	S8	--	Used in CLAM only
Z(98)	S9	--	Not used
Z(99)	S10	--	Not used
Z(100)		g	Mass thrown away (PH2) continuous transport
Z(101)		jerks	Total energy thrown away
Z(102)		g-cm/sh	Total radial momentum thrown away
Z(103)		g-cm/sh	Total axial momentum thrown away
Z(104)		jerks	Energy (internal) added to system when internal is set to 0 if $I < 0$
Z(105)		--	Not used
Z(106)		--	Not used
Z(107)		--	Not used
Z(108)		--	Not used
Z(109)		--	Not used
Z(110)		jerks/g	Critical energy E(S), same value as Z(122)
Z(111)		g/cm^3	Initial density of material
Z(112)		cm/sh	Initial velocity of pellet
Z(113)		--	Epsilonics for emptying pellet $\approx$ .01
Z(114)		--	Not used

Z(115)		g/cm^3	Density (ρ_0)	For equation of state
Z(116)		--	a	
Z(117)		jerks/g	E_0	
Z(118)		--	b	
Z(119)		$jerks/cm^3$	A	
Z(120)		-	V_s	
Z(121)		-	--	
Z(122)		jerks/g	E_s	
Z(123)		--	α	
Z(124)		--	β	
Z(125)		--	--	
Z(126)		$jerks/cm^3$	B	
Z(127)		--	Not used	
Z(128)				
Z(129)				
Z(130)				
Z(131)				
Z(132)				
Z(133)				
Z(134)				
Z(135)				
Z(136)				
Z(137)				
Z(138)		g/cm^3	Density check if $\rho(k) < Z(138)$ stability check for cell (k) is bypassed.	
Z(139)		--	Percent of instability, used in CDT if CABIN $< 0 \approx .5$	
Z(140)		--	Not used	
Z(141)		--		
Z(142)				
Z(143)				
Z(144)				
Z(145)				
Z(146)				
Z(147)		--	j (of pellet-target interface) at t = 0	
Z(148)	A	$10^5 cm/sec$	C(speed of sound = $A + BP^{\epsilon}$ where $A = C_0$ and P is in megabars	
Z(149)	B	--		
Z(150)	ϵ	--		

See Ref. 5 for a more detailed description of the equation of state. For condensed states,

$$P = \left[a + \frac{b}{\frac{E}{E_0\eta^2} + 1}\right]\frac{E}{V} + A\mu + B\mu^2 \quad ;$$

for expanded states,

$$P = aE\rho + \left[\frac{bE\rho}{\frac{E}{E_0\eta^2} + 1} + A\mu e^{-\alpha\left(\frac{V}{V_0} - 1\right)}\right] e^{-\beta\left(\frac{V}{V_0} - 1\right)^2}$$

$$\rho = \frac{1}{V} \qquad \eta = \frac{\rho}{\rho_0} \quad , \text{ and } \quad \mu = \eta - 1 \; .$$

E = specific internal energy

Condensed form for states where

$$\frac{V}{V_0} \leqq 1. \quad \text{and} \quad E < E_S$$

expanded form for states where

$$\frac{V}{V_0} > 1 \quad \text{and} \quad E > E_S$$

The following table contains the constants for the above equations.

EQUATION OF STATE DATA

Loc.	Z(115)	Z(116)	Z(117)	Z(118)	Z(119)	Z(120)	Z(121)	Z(122)	Z(123)	Z(124)	Z(125)	Z(126)
Def.	ρ	a	E_o	b	A	V_s	0	E_s	α	β	0	B
Unit	g/cm^3		jerks/g		jerks/cm^3			jerks/g				jerks/cm^3
W	19.17	.5	2.25 (-5)	1.04	3.08 (-4)	1.11	0.	1.135 (-6)	10.	10.	0.	2.5 (-4)
CU	8.9	.5	3.25 (-5)	1.5	1.39 (-4)	1.13	0.	1.38 (-6)	5.	5.	0.	1.1 (-4)
FE	7.86	.5	9.5 (-6)	1.5	1.279 (-4)	1.21	0.	2.44 (-6)	5.	5.	0.	1.[illegible]5 (-4)
AL	2.7	.5	5. (-6)	1.63	7.52 (-5)	1.1	0.	3. (-6)	5.	5.	0.	6.5 (-5)
BE	1.845	.55	1.75 (-5)	.62	1.173 (-4)	1.1	0.	1. (-5)	5.	5.	0.	5.5 (-5)
Ti	4.51	.5	7. (-6)	.60	1.03 (-4)	1.09	0.	3.5 (-6)	5.	5.	0.	5. (-5)
Ni	8.86	.5	9. (-6)	1.33	1.912	1.11	0.	2.85 (-6)	5.	5.	0.	1.5 (-4)
MO	10.2	.5	4.5 (-6)	1.02	2.713 (-4)	1.08	0.	2.3 (-6)	5.	5.	0.	1.65 (-4)
TH	11.68	.4	2.5 (-6)	.36	5.31 (-5)	1.15	0.	2. (-6)	9.	0.88	0.	5. (-5)
CH_2	.92	.6	7. (-6)	[illegible]	7.5 (-6)	1.1	0.	2.4 (-6)	10.	5.	0.	2. (-6)
PB	11.34	.4	1.5 (-6)	2.[illegible]	4.664 (-5)	1.1	0.	2.6 (-7)	13.	15.	0.	1.5 (-5)

	Z(148)	Z(149)	Z(150)
	C_o	A	ε
AL	5.28	[illegible]	.45
PB	2.0	2.[illegible]	.43
Plastic	2.86	[illegible]	.45
FE	4.04	3.[illegible]	.42
W	4.[illegible]	3.[illegible]	.42

Where

$C = C_o + AP^{\epsilon}$

C_o in 10^5 cm/sec

P in megabars

4. TEST PROBLEMS

4.1. Technique

A series of one-dimensional impact problems was undertaken to select the most appropriate scheme for the velocity weighting in the mass transport equation. An iron projectile traveling at a velocity of 1×10^6 cm/sec striking an iron target was used as the standard test problem. The projectile had five zones of 1-cm each; the target had 95 cones of 1-cm each. The equation of state used was that for iron.(5)

Four different velocity weighting techniques were investigated to determine the best $\bar{u}$ to use in the mass transport expression $\rho\bar{u}\Delta t$. These are illustrated below, for the case where the flow is from cell k to cell (k+1).

The four different approaches were as follows:

1. The donor cell scheme: $\bar{u} = u_{(k)}$

2. The Rich scheme(6) (no density terms) $\bar{u} = u_{(k)} + \frac{u_{(k+1)} - U_{(k-1)}}{4.}$

3. The OIL scheme $$\bar{u} = \frac{\frac{u_{(k)} + u_{(k+1)}}{2.}}{\left[1 + \frac{[u_{(k+1)} - u_{(k)}]\Delta t}{\Delta x}\right]}$$

and 4, another scheme where

$$\bar{u} = \frac{u_{(k)} + u_{(k+1)}}{2.}\left[1 - \frac{[u_{(k+1)} - u_{(k)}]\Delta t}{\Delta x}\right]$$

The donor cell, Fig. 5, looks very good in the neighborhood of the shock front; however, behind the front (the rarefaction side) instability sets in. Rich's scheme (without his density weighting) is very similar to scheme 4. The present scheme 3 now used in the OIL code was chosen as the best representation of the shock front and the rarefaction (Fig. 6). However, the results indicate the possibility of using the donor cell (scheme 1) near the shock front and one of the other three schemes in the rest of the problem.

4.2. Comparison of SHELL and OIL

An iron projectile, 3-cm-diam by 3-cm long, strikes an iron target at a velocity of 4×10^6 cm/sec. The problem was run using both SHELL (PIC transport) and OIL (continuous mass transport.) The starting conditions and cell sizes were the same for both. There were 72 cells in the projectile and 2976 in the target, and 16 particles per cell were used in the SHELL run.

Figure 7 is a plot of the total positive axial momentum and the total positive radial momentum (in the target) vs time for the two problems. Figure 8 shows the shock pressure vs position as a function of time for the two problems. The agreement between the two techniques is very good, with the SHELL scheme overshooting the theoretical shock pressure by a somewhat larger amount. Figures 9 and 10 are mass flux plots of the two schemes vs zone number, again at 45°. Figure 11 is a pressure and compression plot of the two schemes as a function of zone number along a ray 45° from the initial center at the projectile target interface. Figure 12 is the same information at a later time of 9.2 μsec. Throughout all the comparisons, the two schemes are seen to be in very good agreement, with the SHELL scheme exhibiting some undesirable oscillatory behavior due to the discrete nature of the particle population in the cells. Finally, the SHELL run required longer machine time by a factor of 15 over the OIL run.

4.3. One-dimensional Test Problems

Figure 13, a comparison of the two schemes for a plane wave free expansion problem, indicates good agreement between OIL and theory. A hot gas extending to 30 cm with a ρ of .8 g/cm^3, a γ of 5/3, and specific internal energy of 9×10^{-3} jerks/g, with a rigid wall on the left, was suddenly released and free to rarefy to zero pressure. The slight rise in mass in the leading edge is due to the constraint adopted for preventing mass transport to diffuse n cells in n time steps (see Section 3.1).

A second "shock tube" problem consisted of a rigid boundary on the left, a hot gas of ρ = .8 g/cm^3, a γ of 5/3, and a specific internal energy of 9×10^{-3} jerks/g, which was allowed to shock a cold region extending from 30 cm to 40 cm, consisting of a ρ of 1.2 g/cm^3 and a γ of 5/3. In

addition to the OIL and SHELL versions of this flow, a third comparison was made using a plane-geometry version of a one-dimensional Lagrangian (SPUTTER) code. The results are given in Figs. 14 and 15, where pressure and velocity are plotted as functions of distances in both the hot and cold material. The comparison between OIL and SPUTTER is very good, while SHELL, again, exhibits an oscillatory behavior.

Figure 16 shows the initial setup of a spherical blast problem which was run using the three codes, SHELL, OIL, and a spherical version of SPUTTER. Figure 17 is the pressure versus position for the three cases, and Fig. 18 is a plot of density versus position. In the latter plot particularly, SPUTTER displays a decided advantage over the two Eulerian methods. This is due to the fact that the SPUTTER (Lagrangian) zoning was on an equal mass basis and the resolution is accordingly better at big R. Hence, there are many zones to represent the density discontinuity or shock location. The SHELL and OIL results are very similar, with the OIL results being substantially smoother.

Figures 19 and 20 are pressures vs position along the r and z axis for the SHELL and OIL codes. The results for OIL are especially interesting in that they are very nearly spherical, where the results for SHELL are not as smooth and are slightly different along the two axes, partially because of the preferential movement of particles that exist in the PIC transport of SHELL.

The spherical character of the OIL solution is also borne out by further tests, such as the close agreement, to about one-tenth of one percent, of the two components of velocity along a ray 45° from the axis of symmetry. The fact that an initially spherical problem remains spherical is gratifying as evidence that the differencing or transport schemes are not introducing significant preferential treatments in the radial or axial directions.

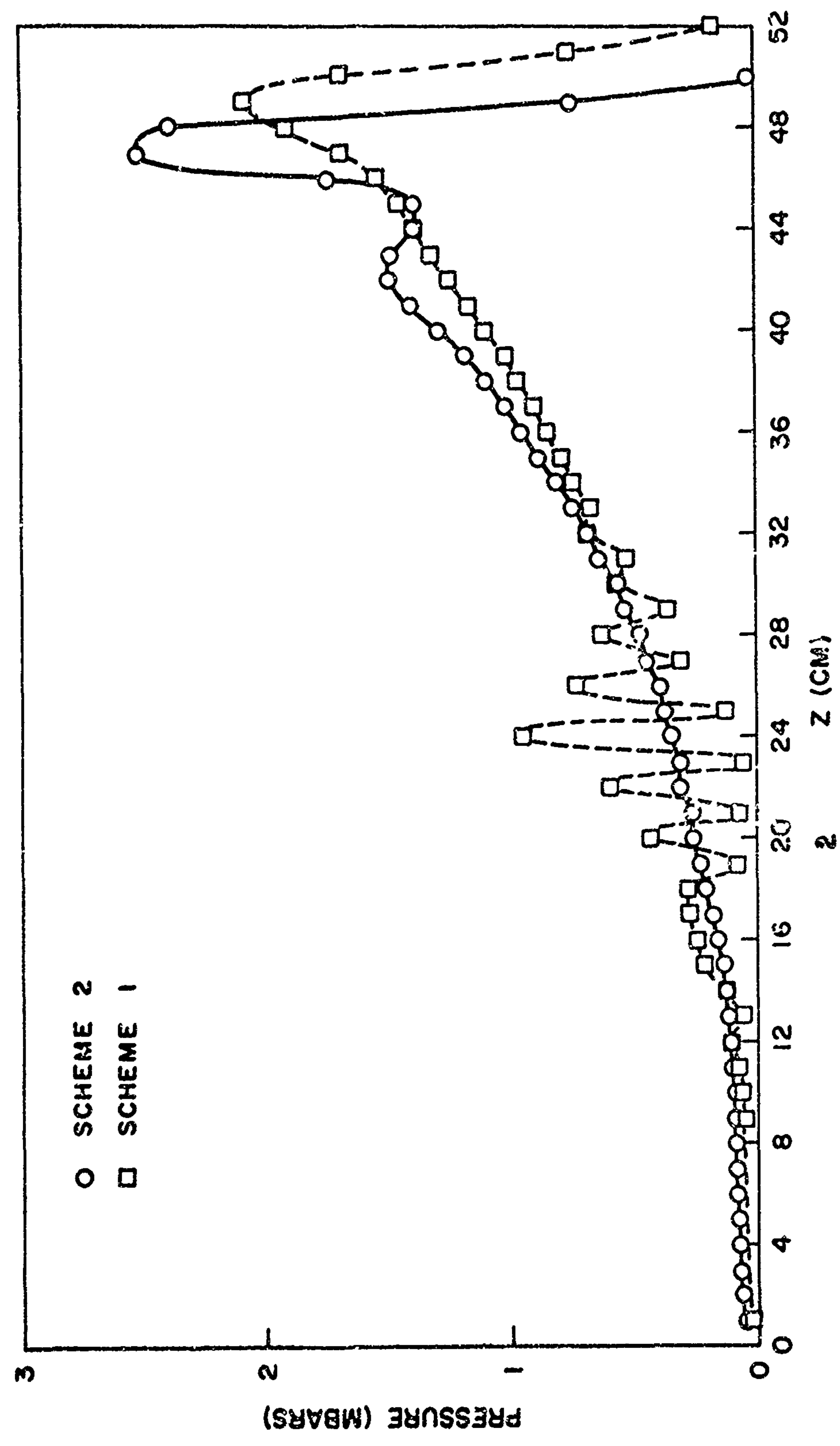

O SCHEME 2
□ SCHEME 1
PRESSURE (MBARS)
Z (CM)
0
1
2
3
0
4
8
12
16
20
24
28
32
36
40
44
48
52

Fig. 5

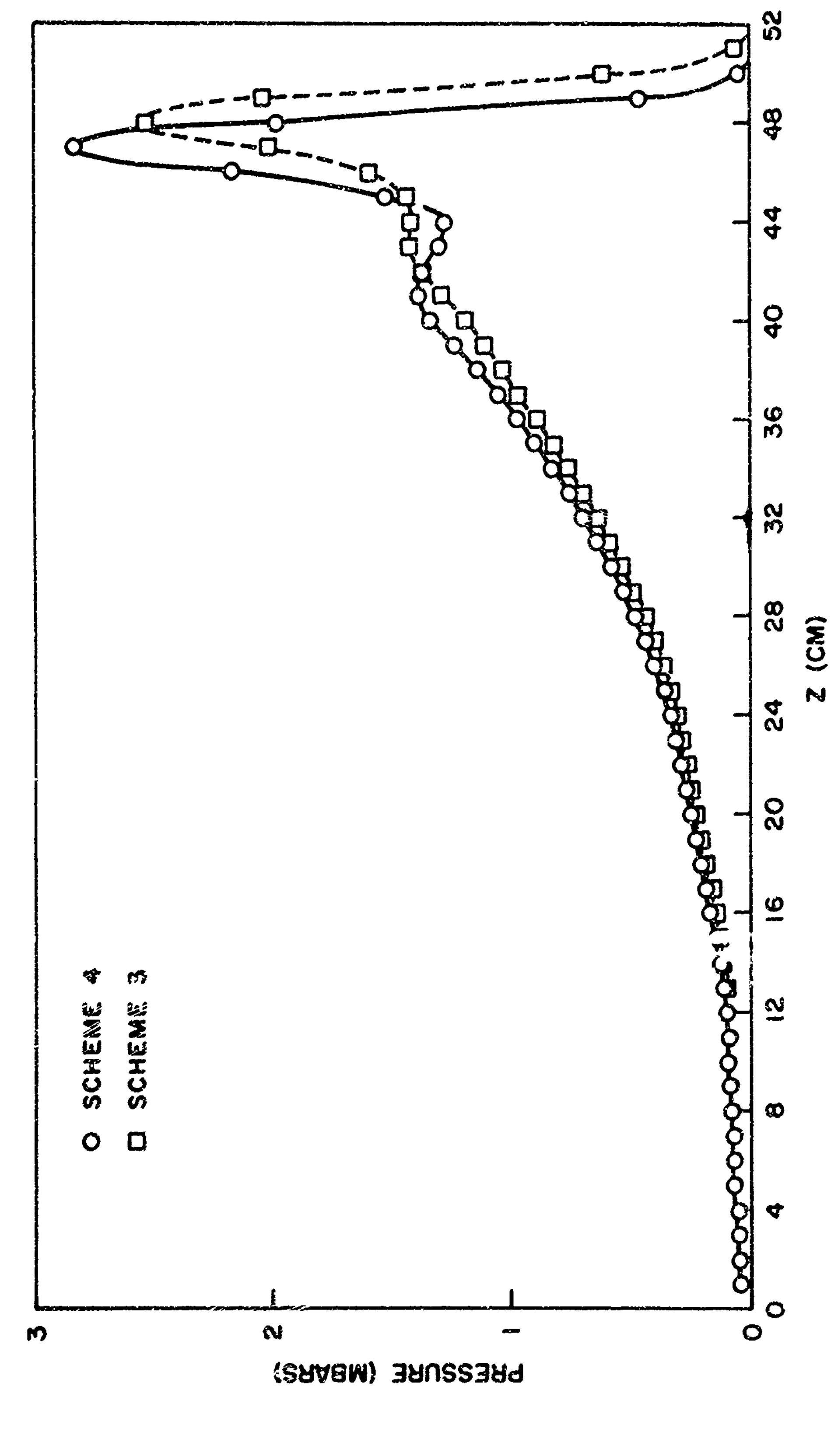


Fig. 6

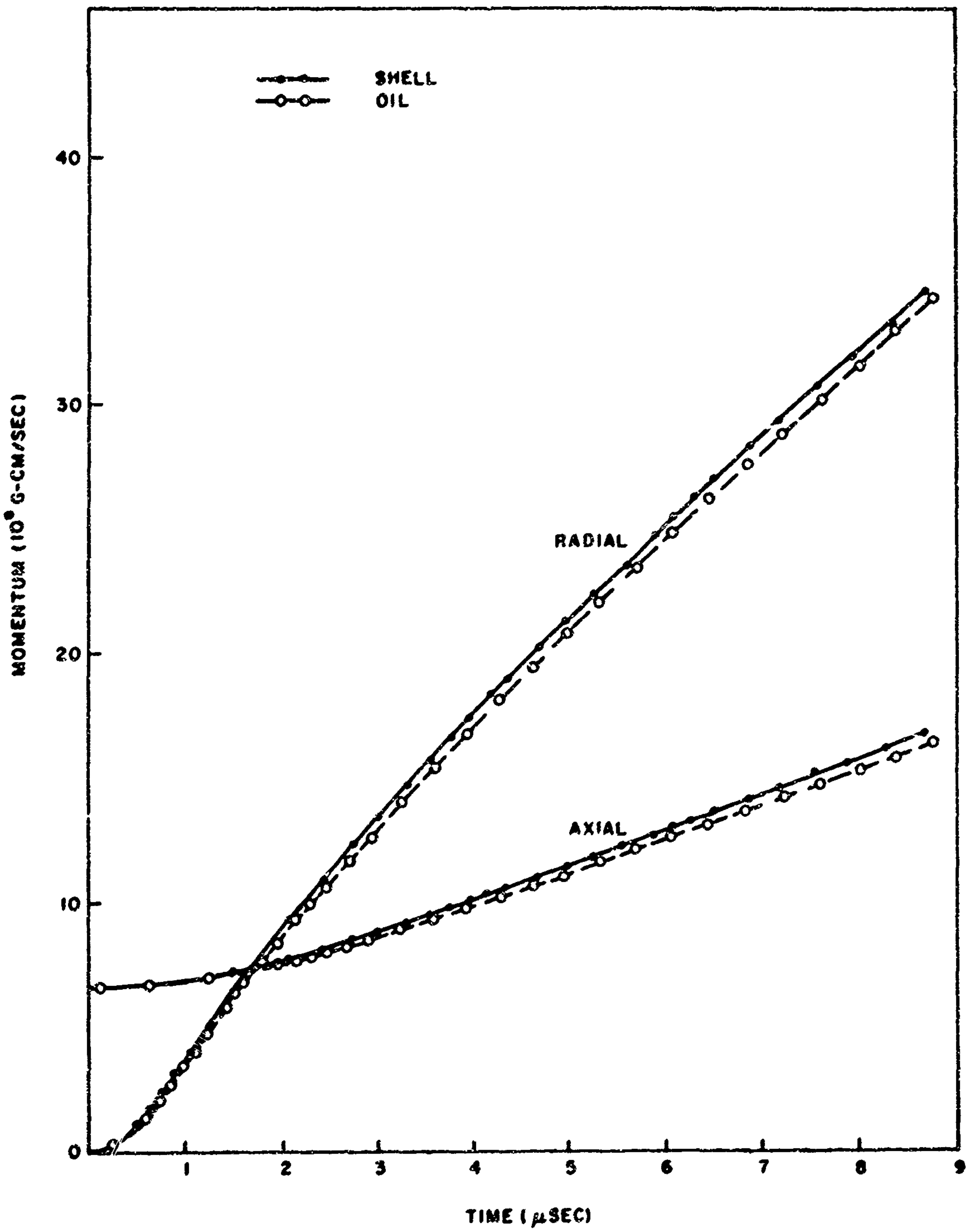

SHELL
OIL
RADIAL
AXIAL
MOMENTUM (10⁸ G-CM/SEC)
0
10
20
30
40
1
2
3
4
5
6
7
8
9
TIME (μSEC)

Fig. 7

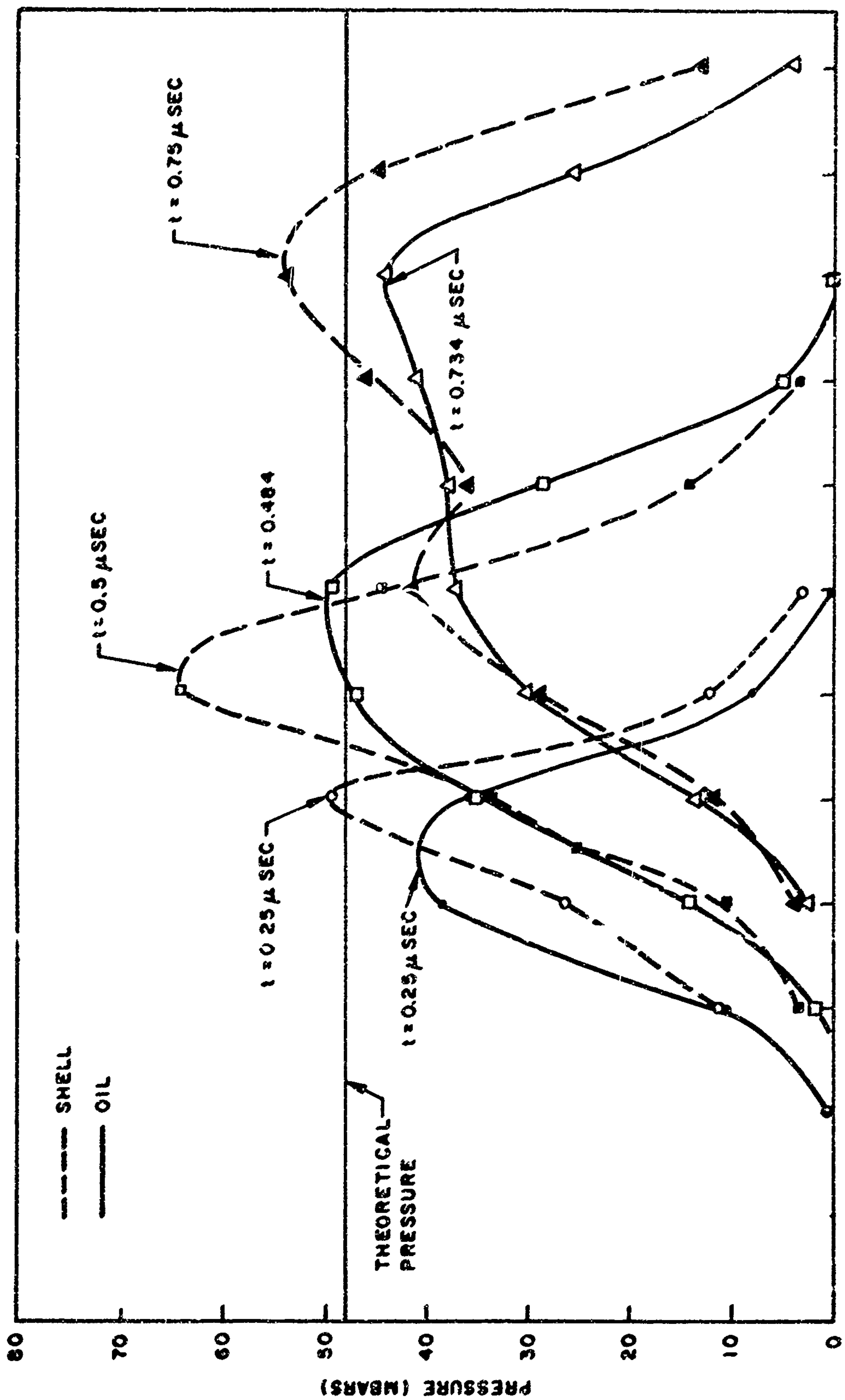

SHELL
OIL
THEORETICAL PRESSURE
t = 0.75 μSEC
t = 0.734 μSEC
t = 0.5 μSEC
t = 0.484
t = 0.25 μSEC
t = 0.25 μSEC
PRESSURE (MBARS)
ZONE NUMBER
0
10
20
30
40
50
60
70
80
16
17
18
19
20
21
22
23
24
25
26
27
28

Fig. 8

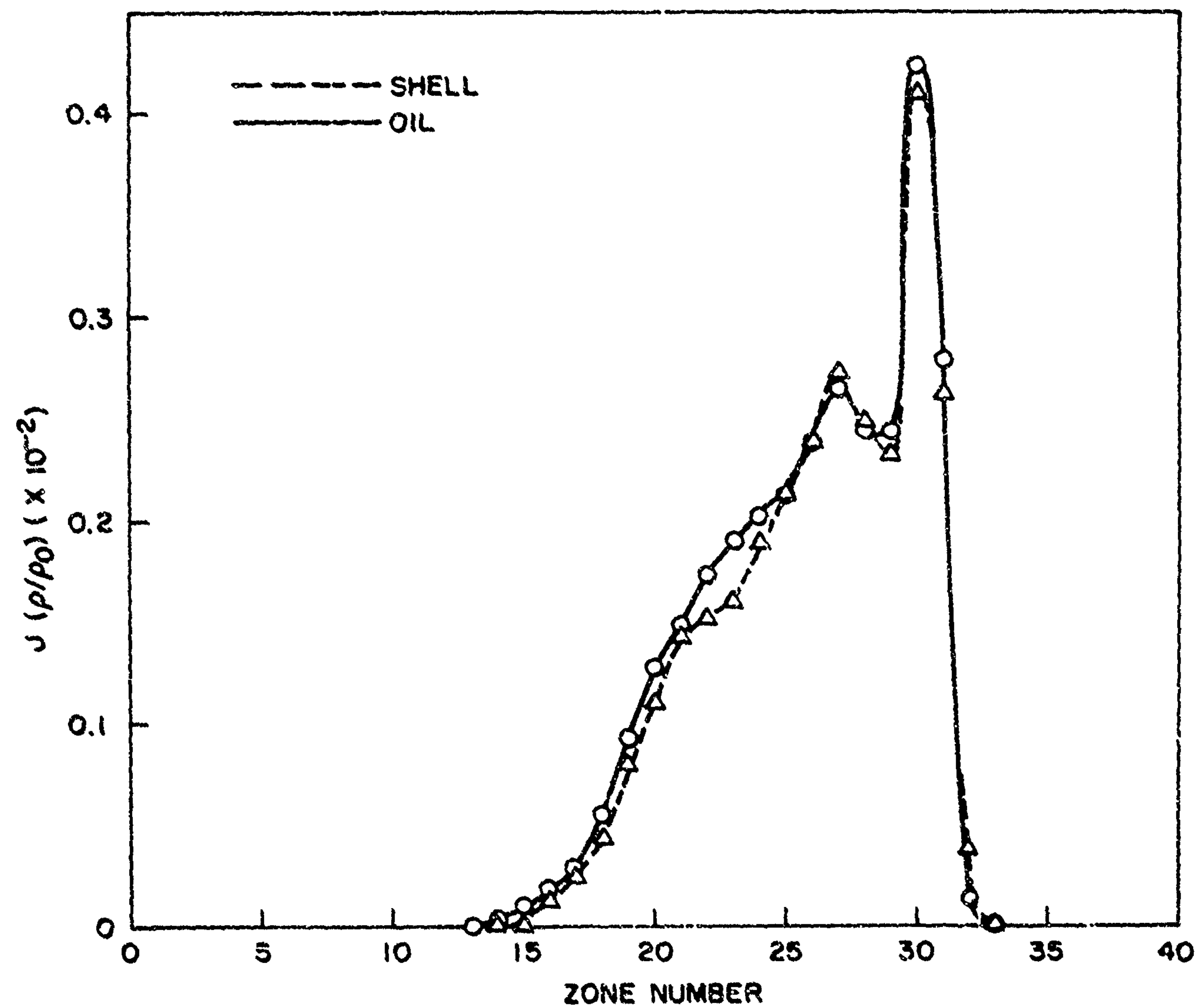

SHELL
OIL
0.4
0.3
0.2
0.1
0
J (ρ/ρ0) (X 10^-2)
0
5
10
15
20
25
30
35
40
ZONE NUMBER

Fig. 9

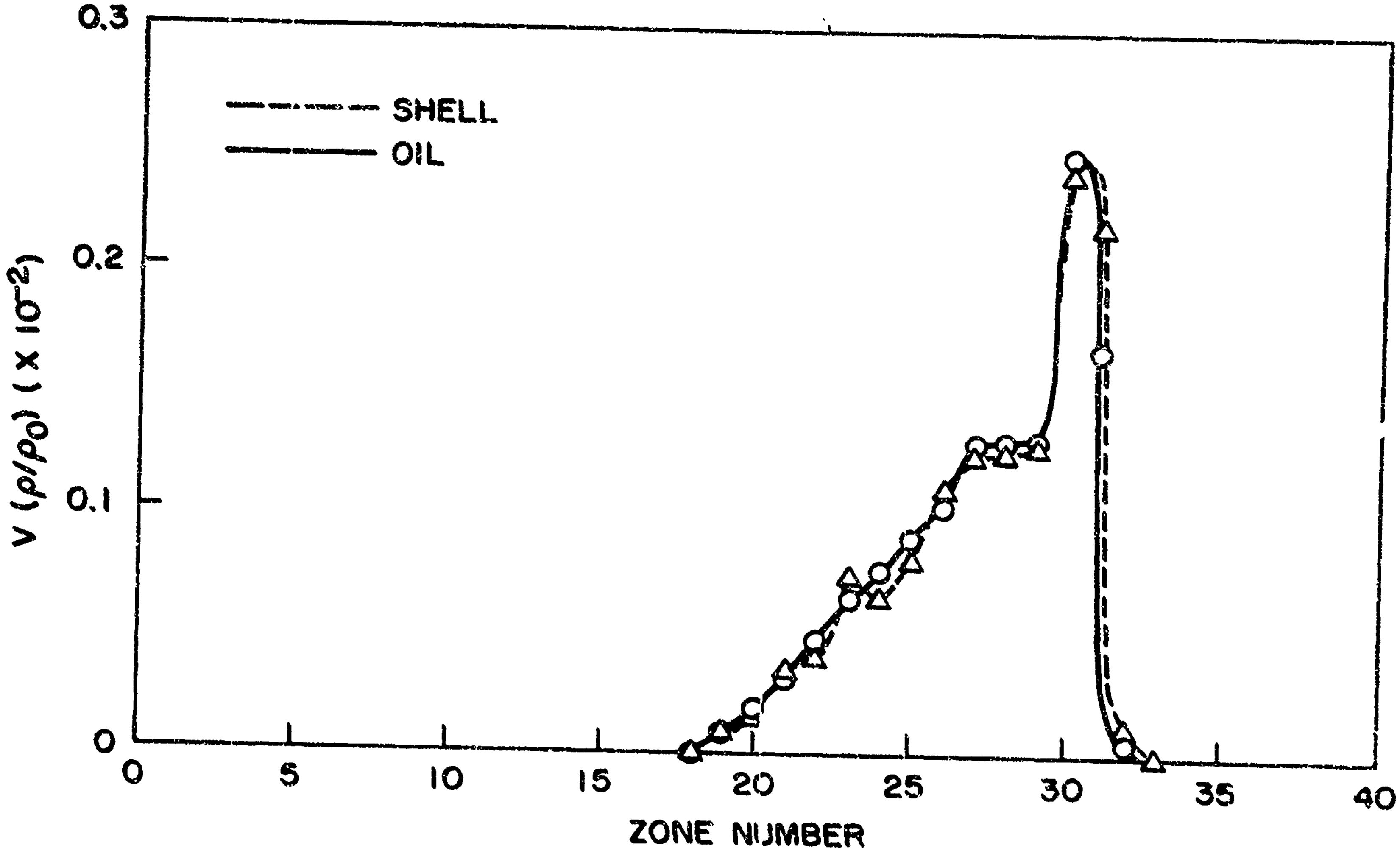

SHELL
OIL
0.3
0.2
0.1
0
$V(\rho/\rho_0)$ (X 10^{-2})
0
5
10
15
20
25
30
35
40
ZONE NUMBER

Fig. 10

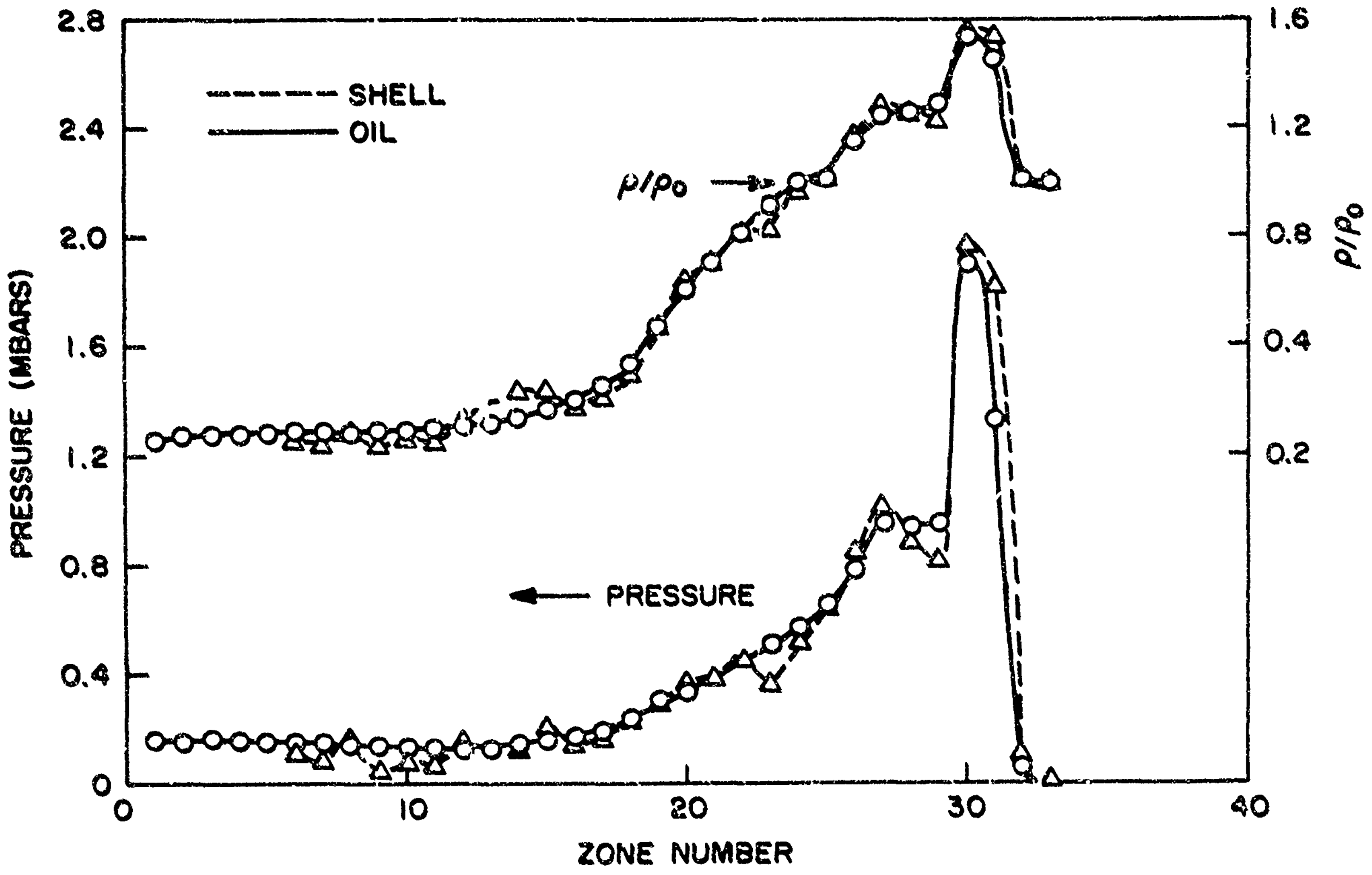

SHELL
OIL
PRESSURE (MBARS)
ZONE NUMBER
PRESSURE
ρ/ρ_0
2.8
2.4
2.0
1.6
1.2
0.8
0.4
0
10
20
30
40
1.6
1.2
0.8
0.4
0.2

Fig. 1ı

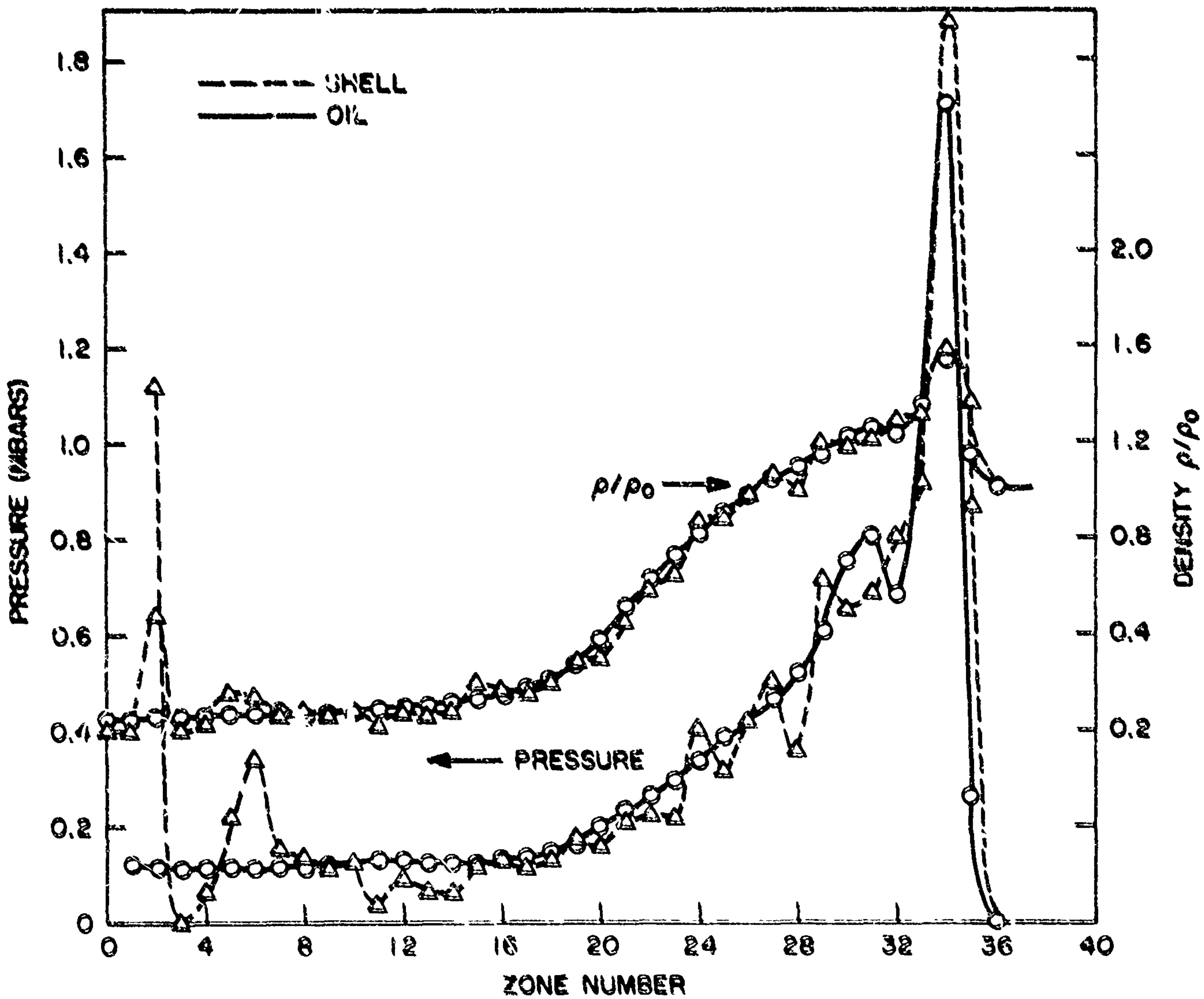
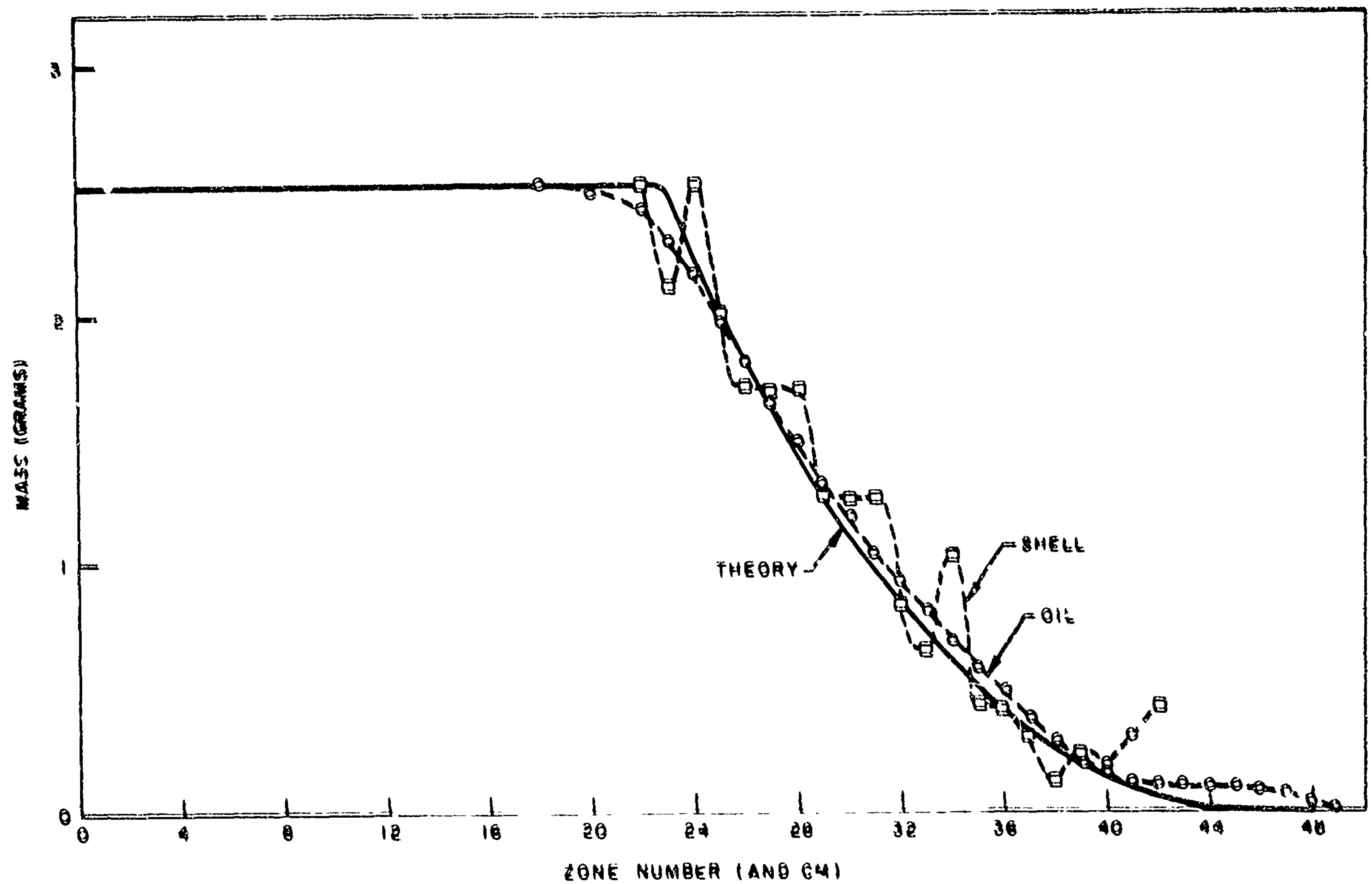
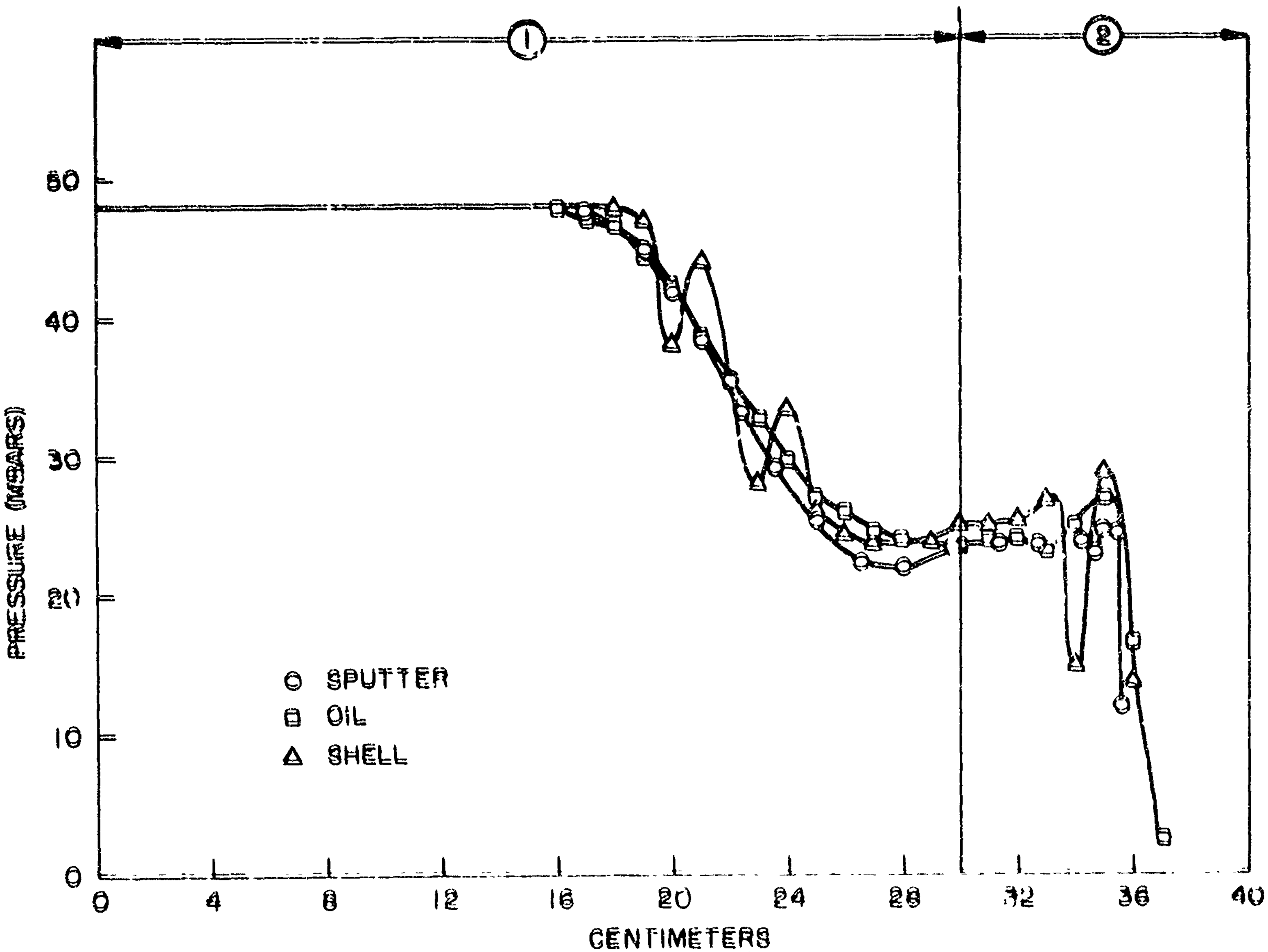
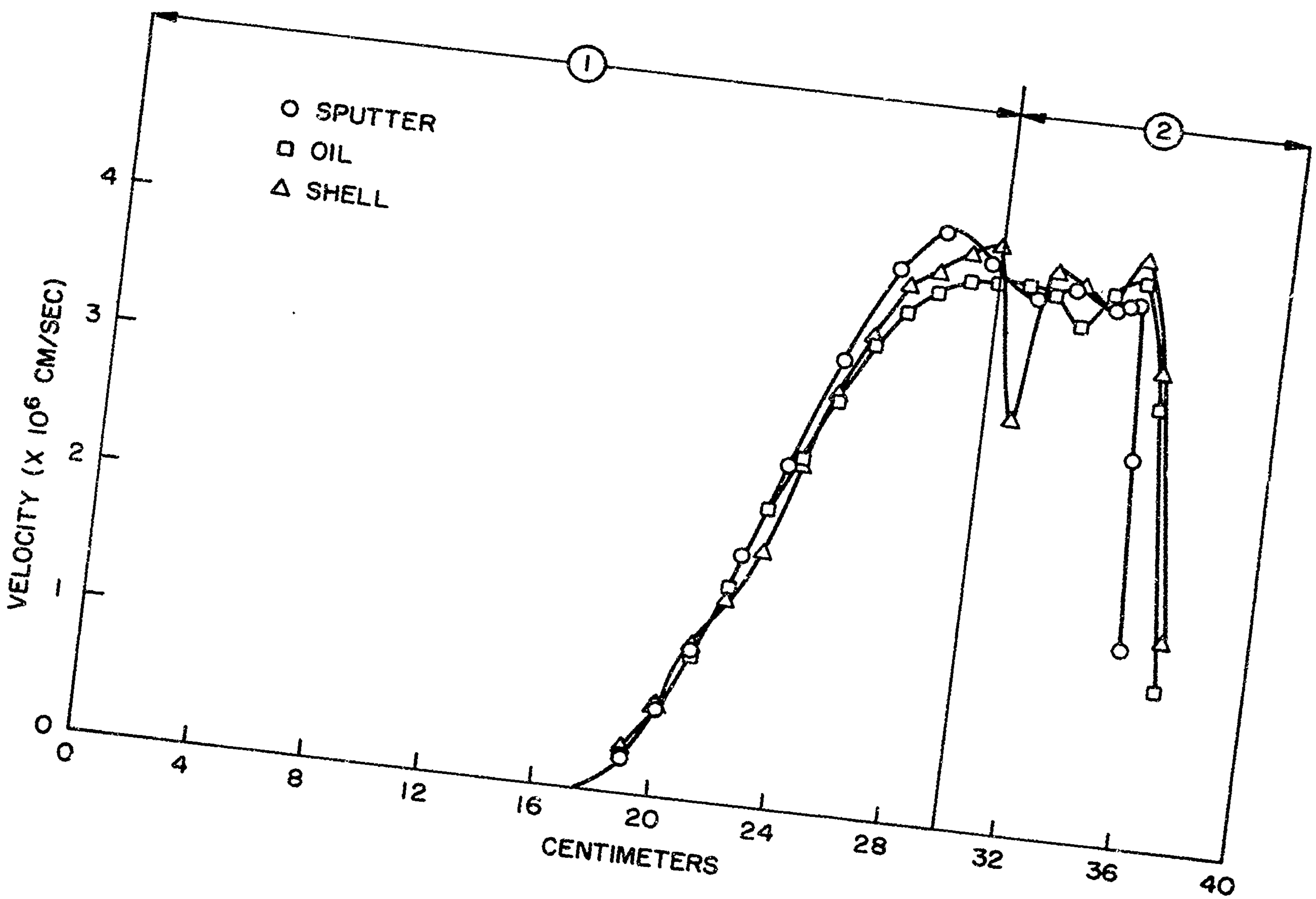

SHELL
OIL
PRESSURE (MBARS)
DENSITY ρ/ρ_0
ρ/ρ_0
PRESSURE
ZONE NUMBER
0
4
8
12
16
20
24
28
32
36
40
0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
0.2
0.4
0.8
1.2
1.6
2.0

Fig. 12



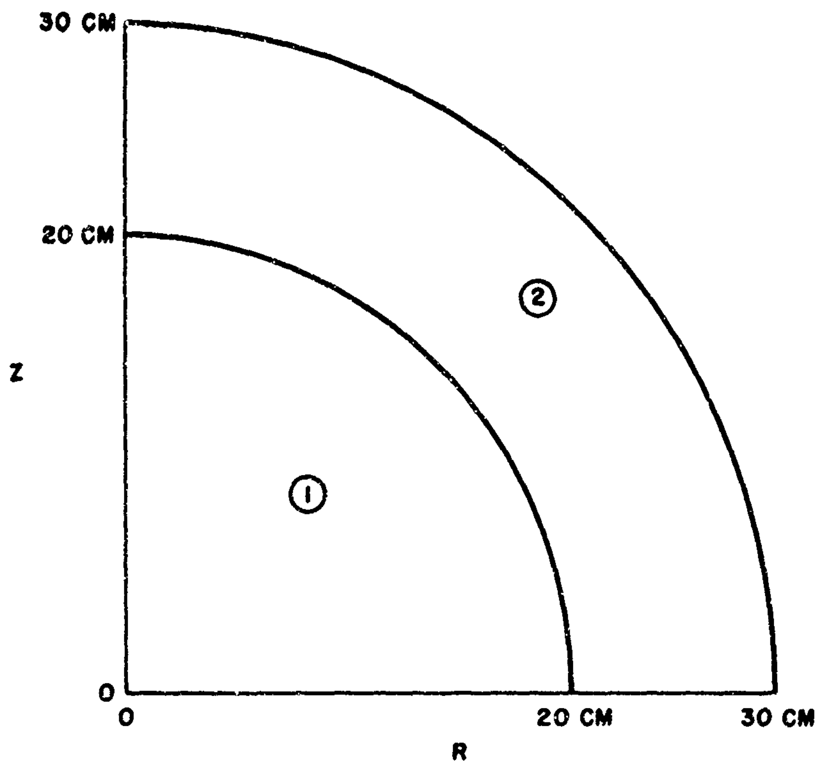
MASS (GRAMS)
ZONE NUMBER (AND CM)
THEORY
SHELL
OIL


1
2
PRESSURE (MBARS)
50
40
30
20
10
0
0
4
8
12
16
20
24
28
32
36
40
CENTIMETERS
SPUTTER
OIL
SHELL



SPUTTER
OIL
SHELL
1
2
VELOCITY (X 10^6 CM/SEC)
CENTIMETERS
0
4
8
12
16
20
24
28
32
36
40
0
1
2
3
4

Fig. 15



REGION ① ρ = 0.8, γ = 5/3

I = SPECIFIC INTERNAL ENERGY

= 9 X 10^{13} ERGS

U = V = 0

REGION ② ρ = 1.2, γ = 5/3

I = U = V = 0

Fig. 16

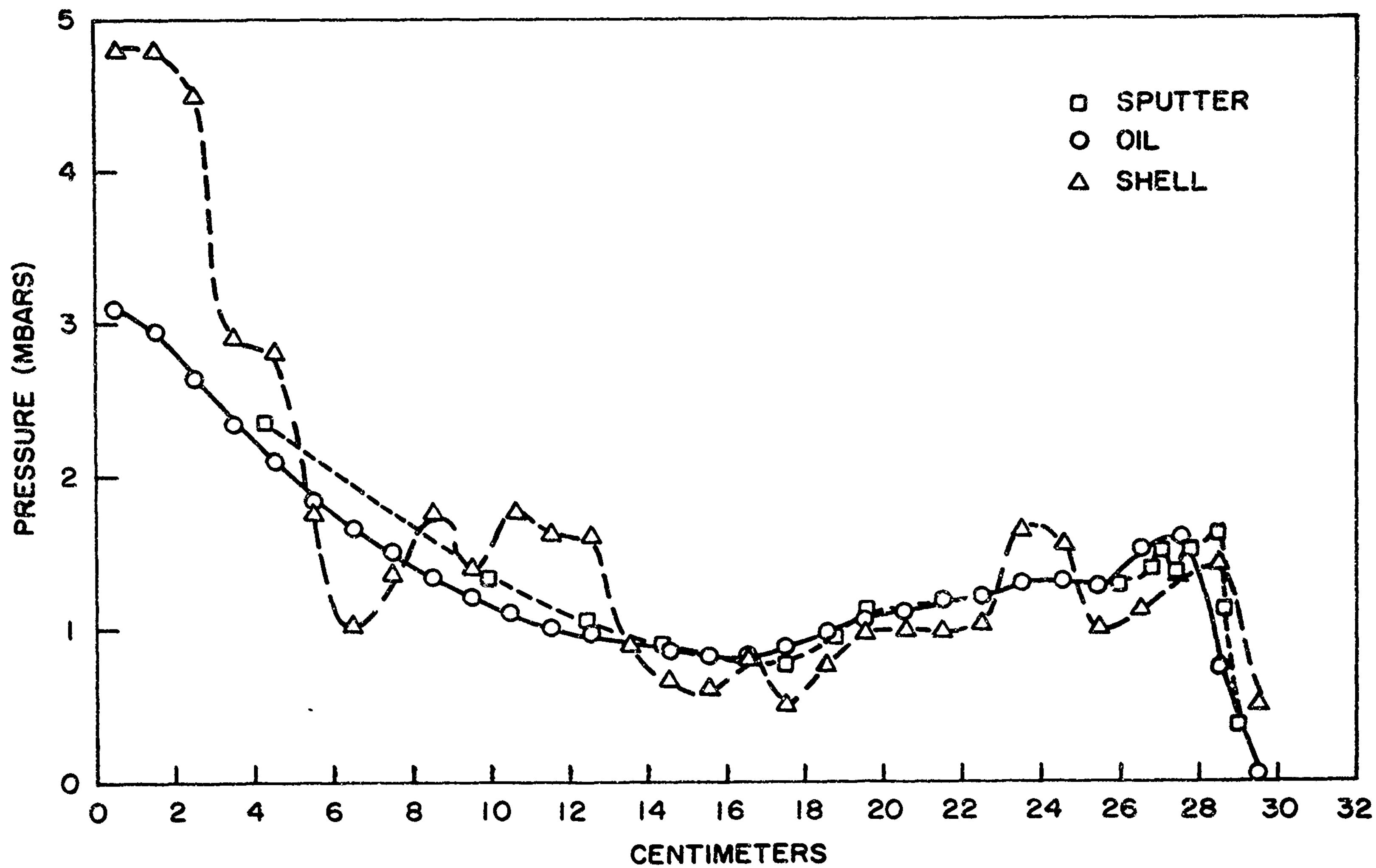
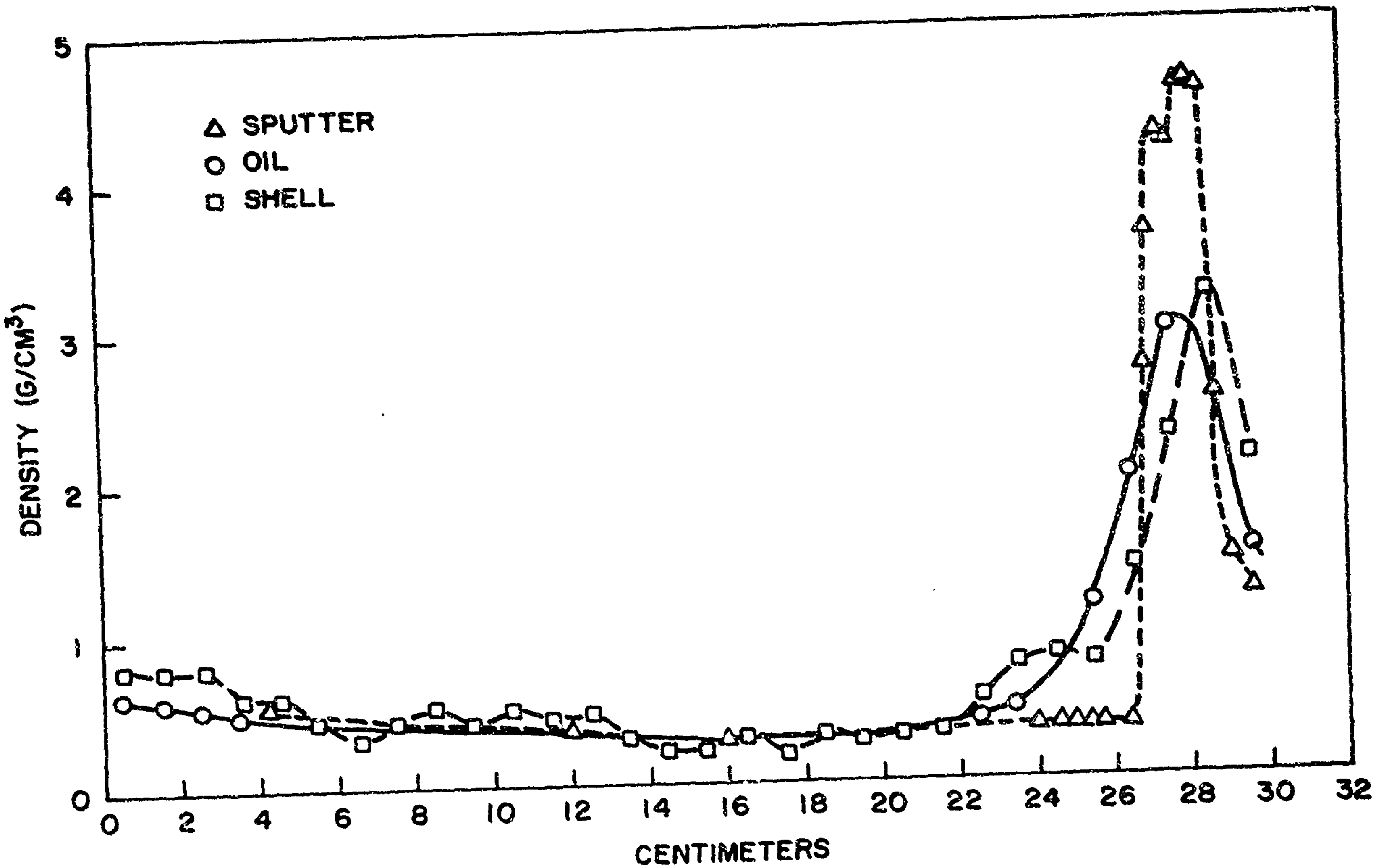

SPUTTER
OIL
SHELL
PRESSURE (MBARS)
CENTIMETERS
0
1
2
3
4
5
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32

Fig. 17



SPUTTER
OIL
SHELL
DENSITY (G/CM3)
CENTIMETERS
0
1
2
3
4
5
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32

Fig. 18

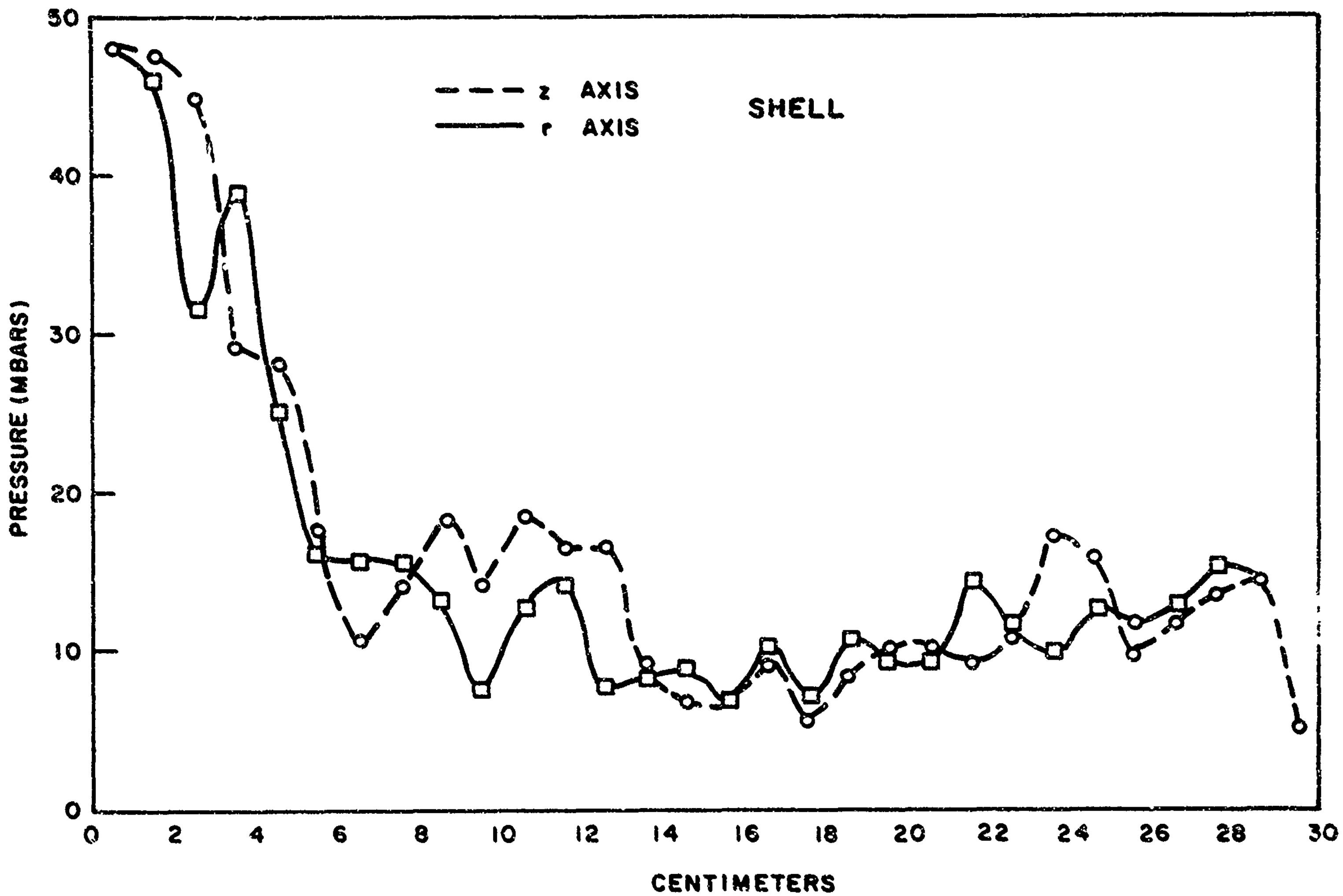
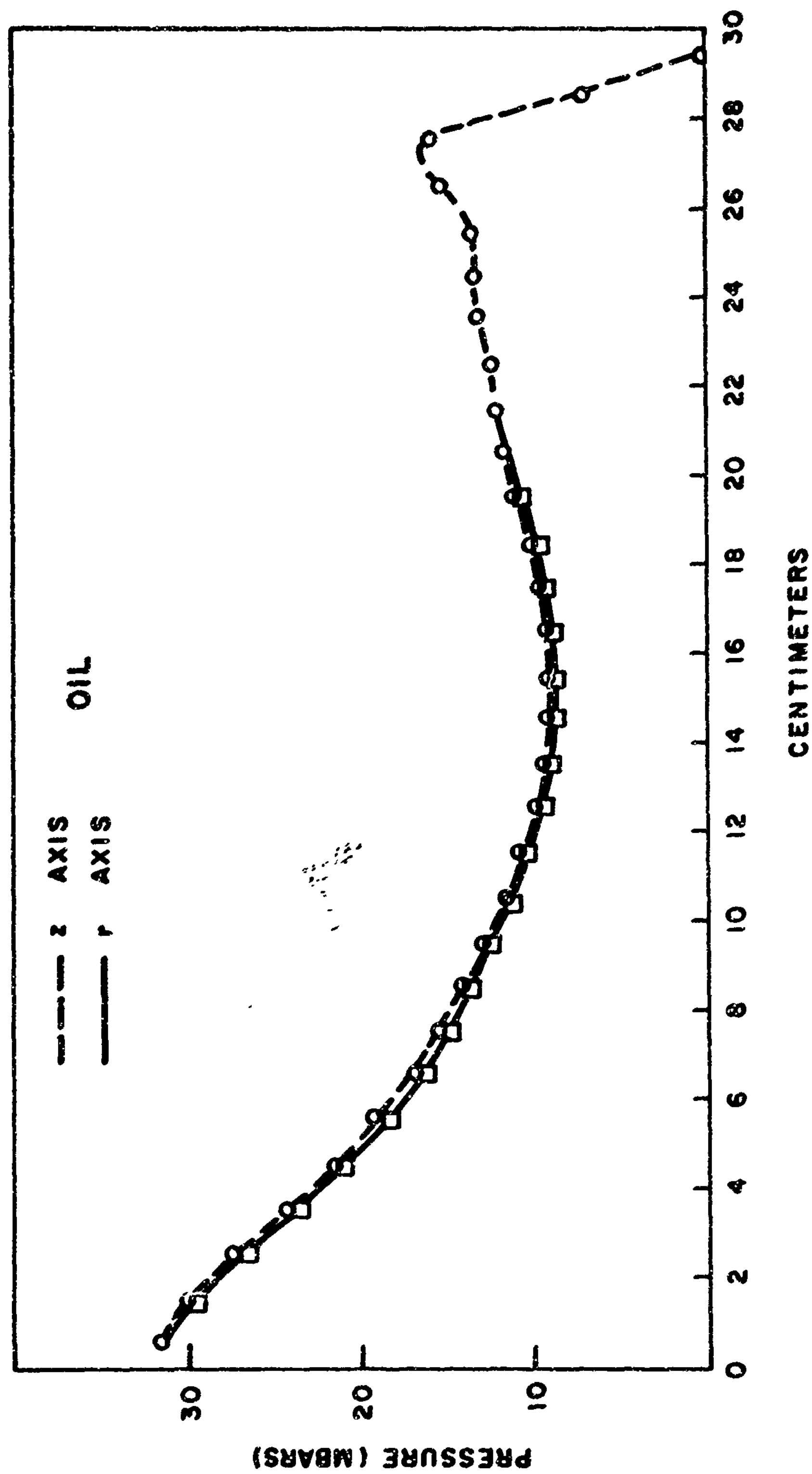


Fig. 19



z AXIS
r AXIS
OIL
PRESSURE (MBARS)
30
20
10
CENTIMETERS
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30

Fig. 20

```
C       NOTE, THE BELOW SET OF
C       DIMENSION, EQUIVALENCE AND COMMON IS
C       TO BE USED WITH ALL SUBROUTINES IN
C       CLAM, WITH THE EXCEPTION OF MAIN ROUTINE.
C         D    I    M    E    N    S    I    O    N                               INPU0020
C                                                                                 INPU0030
      DIMENSION AIX(4300),AM(130),                                                INPU0040
     1AMX(4300),DX(52),ENDD(2),ITAB(502),IZ(100),                                 INPU0050
     2RONE(2),TAB(502),TABI(20),TABIY(21),TABR(20),                               INPU0060
     3TABUV(20),TABX(21),TABY(21),TAU(52),TEMP(13),                               INPU0070
     4U(4300),V(4300),X(53),XL(130),XX(54),                                       INPU0080
     5YL(130),Z(150),Y(100),YY(101),DY(100)                                       INPU0090
      DIMENSION IW1(130),IW2(130)                                                 INPU0100
      COMMON          Z       ,XX      ,TAB     ,YY                               INPU0110
      COMMON          AID     ,AIMAX   ,AIX     ,AJMAX   ,AM      ,AMD            INPU0120
      COMMON          AMX     ,DX      ,ENDD    ,FMX     ,GXN     ,GXX            INPU0130
      COMMON          GYN     ,GYX     ,I       ,IA      ,IB      ,IBA            INPU0140
      COMMON          IBB     ,ID      ,IG      ,II      ,IIC     ,IJ             INPU0150
      COMMON          IR      ,IRC     ,IUV     ,IUVC    ,IWS     ,IWSA           INPU0160
      COMMON          IWSB    ,IX      ,IXN     ,IXX     ,IYN     ,IYX            INPU0170
      COMMON          J       ,JA      ,JI      ,JIM     ,K       ,KE             INPU0180
      COMMON          KF      ,KK      ,L       ,LA      ,LB      ,LD             INPU0190
      COMMON          LE      ,LI      ,LX      ,M       ,MI      ,MIJ            INPU0200
      COMMON          MJ      ,MN      ,MNP     ,MX      ,MXA     ,MXS            INPU0210
      COMMON          MZ      ,NPKS    ,NPP     ,NT      ,NX      ,NY             INPU0220
      COMMON          GOOCFL,RHO    ,RONE,SLA,SLB    ,TABI                        INPU0230
      COMMON          TABIY   ,TABR    ,TABUV   ,TABX    ,TABY    ,TAM            INPU0240
      COMMON          TAU     ,TEMP    ,TFMX    ,TPIDY   ,TX      ,TY             INPU0250
      COMMON          U       ,V       ,NPIDY   ,WS      ,WSA     ,WSB            INPU0260
      COMMON          WSC     ,WSD     ,WSE     ,WSF     ,WSG     ,WSI            INPU0270
      COMMON          WSL     ,WSU     ,WSV     ,WSX     ,WSY     ,WS5            INPU0280
      COMMON          XC      ,XL      ,YC      ,YL      ,YMAX    ,WSR            INPU0290
      COMMON          PE      ,PM      ,TTX     ,TTY     ,LF      ,E              INPU0300
      COMMON          PEE     ,NPRR    ,NYY     ,DY      ,NK      ,SWITCH         INPU0310
      COMMON          IW1     ,IW2                                                INPU0320
C                                                                                 INPU0400
C                E   Q   U   I   V   A   L   E   N   C   E                        INPU0410
C                                                                                 INPU0420
     0EQUIVALENCE       (Z,IZ,PROB),       (Z(2),CYCLE),       (Z(3),DT),        INPU0430
     1(Z(4),PRINTS),    (Z(5),PRINTL),     (Z(6),DUMPT7),      (Z(7),CSTOP),     INPU0440
     2(Z(8),PIDY),      (Z(9),TMZ),        (Z(10),GAM),        (Z(11),GAMD),     INPU0450
     3(Z(12),GAMX),     (Z(13),ETH),       (Z(14),FFA),        (Z(15),FFB),      INPU0460
     4(Z(16),TMDZ),     (Z(17),TMXZ),      (Z(18),XMAX),       (Z(19),TXMAX),    INPU0470
     5(Z(20),TYMAX),    (Z(21),AMDM),      (Z(22),AMXM),       (Z(23),DNN),      INPU0480
     6(Z(24),DMIN),     (Z(25),FEF),       (Z(26),DTNA),       (Z(27),CVIS),     INPU0490
     7(Z(28),NPR),      (Z(29),NPRI),      (Z(30),NC),         (Z(31),NPC),      INPU0500
     8(Z(32),NRC),      (Z(33),IMAX),      (Z(34),IMAXA),      (Z(35),JMAX),     INPU0510
     9(Z(36),JMAXA),    (Z(37),KMAX),      (Z(38),KMAXA),      (Z(39),NMAX)      INPU0520
     0EQUIVALENCE       (Z(40),ND),        (Z(41),KDT),        (Z(42),IXMAX),    INPU0530
     1(Z(43),NOD),      (Z(44),NUPR),      (Z(45),NIMAX),      (Z(46),NJMAX),    INPU0540
```

```
     2(Z(47),I1),       (Z(48),I2),       (Z(49),I3),       (Z(50),I4),      INPU0550
     3(Z(51),N1),       (Z(52),N2),       (Z(53),N3),       (Z(54),N4),      INPU0560
     4(Z(55),N5),       (Z(56),N6),       (Z(57),N7),       (Z(58),N8),      INPU0570
     5(Z(59),N9),       (Z(60),N10),      (Z(61),N11),      (Z(62),NRM),     INPU0580
     6(Z(63),TRAD),     (Z(64),XNRG),     (Z(65),SN),       (Z(66),DXN),     INPU0590
     7(Z(67),RADER),    (Z(68),RADET),    (Z(69),RADEB),    (Z(70),DTRAD),   INPU0600
     8(Z(71),REZFCT),   (Z(72),RSTOP),    (Z(73),SHELL),    (Z(74),BBOUND),INPU0610
     9(Z(75),TOZONE),   (Z(76),ECK),      (Z(77),SBOUND),   (Z(78),X1)       INPU0620
     0EQUIVALENCE       (Z(79),X2),       (Z(80),Y1),       (Z(81),Y2),      INPU0630
     1(Z(82),CABLN),    (Z(83),VISC),     (Z(84),T),        (Z(85),GMAX),    INPU0640
     2(Z(86),WSGD),     (Z(87),WSGX),     (Z(88),GMADR),    (Z(89),GMAXR),   INPU0650
     3(Z(90),S1),       (Z(91),S2),       (Z(92),S3),       (Z(93),S4),      INPU0660
     4(Z(94),S5),       (Z(95),S6),       (Z(96),S7),       (Z(97),S8),      INPU0670
     5(Z(98),S9),       (Z(99),S10)                                          INPU0680
     0EQUIVALENCE       (Z,IZ),           (XX(2),X(1)),     (TAB,ITAB),      INPU0690
     1(YY(2),Y(1))                                                           INPU0700
C                                                                            INPU0710
     NOTE . ALTHOUGH THE DIMENSIONS FOR THE CELL
     QUANTITIES ARE  4300 IN  CLAM, THE DIMENSIONS
     FOR  OIL ARE 3500, THAT IS, KEEP (IMAX)(JMAX)
     +1 LESS THAN  3499.
```

```
$IBFTC MAIN    LIST,DECK,REF
CMAIN                                                                MAIN0010
C L A M                   ******   M A I N   ******                  MAIN0020
C                                                                    MAIN0030
C     ** IF PROBLEM NUMBER IS NEGATIVE,
C     CLAM WILL WRITE THE PARTICLES ON TAPE,
C     PREPARING IT FOR A PIC RUN.
C     *** NOTE (1 MATERIAL ONLY (1X))                                MAIN0040
C                                                                    MAIN0050
      CALL SLITE (0)                                                 MAIN0060
C     INPUT ROUTINE CALCULATES THE ACTUAL GRID,
C     DIMENSIONS AND INDICES.
   10 CALL INPUT                                                     MAIN0070
C     PH1, READS IN DATA CARDS FOR THE
C     PACKAGES, PH2 CALCULATES THE GEOMETRICS,
C     PH3 THE PARTICLES, PH4 CALLS THE
C     6 POSSIBLE FITS THAT CALCULATE THE
C     DENSITY, VELOCITIES AND INTERNAL ENERGY
C     OF THE PARTICLES.
   20 CALL PH1                                                       MAIN0080
C     OUTPUT CALCULATES THE VELOCITY (BOTH
C     RADIAL AND AXIAL) AND SPECIFIC INTERNAL
C     ENERGY OF EACH CELL FROM THE
C     TOTAL MOMENTA AND INTERNAL
C     ENERGY AND MASS OF EACH CELL.
C     OUTPUT ALSO PREPARES A DUMP TAPE
C     WHICH IS USED THEN TO START OIL.
   30 CALL OUTPUT                                                    MAIN0090
      CALL EXIT                                                      MAIN0100
      END                                                            MAIN0110
```

```
$IBFTC INPUT   LIST,DECK,REF
      SUBROUTINE INPUT                                                  INPU0010
C                                                                       INPU0730
C                                                                       INPU0940
C     ****** NOTE (1 MATERIAL ONLY ((X))                                INPU0950
      MZ=150                                                            INPU0960
C     CLEAR Z BLOCK.
      DO 30 I=1,MZ                                                      INPU0970
   30 Z(I)=0.0                                                          INPU0980
C                 READ IN HEADING CARD                                  INPU0990
      READ  (5,8012)IWS                                                 INPU1000
      IWS=1                                                             INPU1010
      WRITE (6,8012)(IWS)                                               INPU1020
      WRITE (6,8100)                                                    INPU1030
C                 READ IN PROBLEM CONSTANTS                             INPU1040
C     PROB=PROBLEM NO. AIMAX=IMAX,
C     AJMAX=JMAX, QCOOFL IS NOT USED-SET
C     TO ZERO, SHELL SET=2.,S8,S9 ARE
C     ZERO, SET N7 TO=TAPE NO.
      READ  (5,8004)PROB,AIMAX,AJMAX,QOOOFL,SHELL,S8,S9,N7              INPU1050
      IF(N7)40,40,50                                                    INPU1060
   40 N7=7
   50 CONTINUE                                                          INPU1080
C     MAX. NUMBER OF ZONES IN R DIRECTION.
      MI=50                                                             INPU1090
C     MAX. NUMBER OF ZONES IN Z DIRECTION.
      MJ=100                                                            INPU1100
C     MAX. NUMBER OF PARTICLES/CELL.
      MNP=400                                                           INPU1110
C                 SIZE OF TABLE (TAB)                                   INPU1120
      JTM=500                                                           INPU1130
C                 MAXIMUM  I*J                                          INPU1140
C     MAX. NUMBER OF CELLS.
   60 MIJ=4299                                                          INPU1150
C     CALCULATE ADDITIONAL INDICES FOR CLAM AND OIL.
   70 IMAX=AIMAX                                                        INPU1160
      JMAX=AJMAX                                                        INPU1170
      IMAXA=IMAX+1                                                      INPU1180
      IXMAX=IMAXA+1                                                     INPU1190
      JMAXA=JMAX+1                                                      INPU1200
      KMAX=(IMAX*JMAX)+1                                                INPU1210
      KMAXA=KMAX-1                                                      INPU1220
      WRITE (6,6048)(PROB,IMAX,JMAX)                                    INPU1230
C     CHECK INPUT NOS. CONCERNED WITH GRID SIZE.
  101 IF(IMAX-MI)102,102,9901                                           INPU1240
  102 IF(JMAX-MJ)104,104,9902                                           INPU1250
  104 IF(KMAX-MIJ-1)106,106,9903                                        INPU1260
  106 NOD=1                                                             INPU1270
      NPC=1                                                             INPU1280
      NRC=0                                                             INPU1290
```

```
C                 READ IN DY AND DX
      I=0
      J=0
      X(I)=0.0
      Y(J)=0.0
 2000 READ  (5,8102)IWSA,IWSB,N1,N2,N3,N4,(TEMP(K),K=1,4)
      L=1
C     COUNT NO. OF DIFFERENT DX OR DY.
      IF(N4)2003,2001,2003
 2001 IF(N3)2004,2002,2004
 2002 IF(N2)2006,2008,2006
 2003 L=L+1
 2004 L=L+1
 2006 L=L+1
 2008 IF(IWSB)2010,2010,2030
C     PROCESS THE DX AND DY VALUES.
 2010 DO 2014 N=1,L
      NK=IZ(N+50)
      DO 2012 K=1,NK
      I=I+1
      DX(I)=TEMP(N)
      X(I)=X(I-1)+DX(I)
 2012 CONTINUE
 2014 CONTINUE
      GO TO 2050
C     CALC THE  Y  AND  DY  VALUES
 2030 DO 2034 N=1,L
      NK=IZ(N+50)
      DO 2032 K=1,NK
      J=J+1
      DY(J)=TEMP(N)
      Y(J)=Y(J-1)+DY(J)
 2032 CONTINUE
 2034 CONTINUE
 2050 IF(IWSA)2052,2000,2052
C     IF(=) READ MORE DX OR DY DATA CARDS.
 2052 IF(J-JMAX)9905,2053,9905
C     CHECK INPUT NUMBERS.
 2053 IF(I-IMAX)9906,2054,9906
 2054 CONTINUE
      READ  (5,8004)WS,WSA,WSB,SWITCH
C     N1, AND N2 ARE THE 2 SCRATCH TAPES.
      N1=WS
      N2=WSA
      REWIND N1
      REWIND N2
C     N4=MAX. NUMBER OF PARTICLES-1 PER RECORD.
      N4=WSB
      NPRI=N4
      NPRR=N4
```

```
      WRITE (6,8064)IMAX,(X(I),I=0,IMAX)                                INPU1740
      WRITE (6,8065)JMAX,(Y(J),J=0,JMAX)                                INPU1750
      WS=3.1415927                                                      INPU1760
      WSA=0.0                                                           INPU1770
C     CALCULATE THE AREA*S(TAU)=PI(R(I)**2-
C     R(I-1)**2).
      DO 1008 I=1,IMAX                                                  INPU1780
      WSB=WSA                                                           INPU1790
      WSA=X(I)**2                                                       INPU1800
 1008 TAU(I)=WS*(WSA-WSB)                                               INPU1810
C     WRITE OUT X,Y,DX,DY, AND TAU VALUES.
      WRITE (6,8066)IMAX,(DX(I),I=1,IMAX)                               INPU1820
      WRITE (6,8067)JMAX,(DY(I),I=1,JMAX)                               INPU1830
      WRITE (6,8092)(IMAX,(TAU(I),I=1,IMAX))                            INPU1840
 1010 XMAX=X(IMAX)                                                      INPU1850
      TXMAX=XMAX*2.0                                                    INPU1860
      YMAX=Y(JMAX)                                                      INPU1870
      TYMAX=YMAX*2.0                                                    INPU1880
C     PIDY IS REALLY PI(3.1415927).
      PIDY=WS                                                           INPU1890
C     SET VELOCITIES, INTERNAL ENERGIES AND MASSES
C     TO 0.
      DO 1014 I=1,KMAXA                                                 INPU1910
      U(I)=0.0                                                          INPU1920
      V(I)=0.0                                                          INPU1930
      AIX(I)=0.0                                                        INPU1940
      AMX(I)=0.0                                                        INPU1950
 1014 CONTINUE                                                          INPU1960
C     SET TOTAL ENERGY TO ZERO.
      ETH=0.0                                                           INPU1970
C     INITIALIZE MIN. MASS PARTICLE TO A LARGE NO.
      AMDM=1.E+28                                                       INPU1980
      AMXM=AMDM                                                         INPU1990
      GO TO 2016                                                        INPU2000
C                ERROR                                                  INPU2010
 9901 NK=101                                                            INPU2020
      GO TO 9999                                                        INPU2030
 9902 NK=102                                                            INPU2040
      GO TO 9999                                                        INPU2050
 9903 NK=104                                                            INPU2060
      GO TO 9999                                                        INPU2070
C                JMAX DOES NOT EQUAL THE SUM OF THE INPUT J             INPU2080
 9905 NK=2052                                                           INPU2090
      GO TO 9999                                                        INPU2100
C                IMAX DOES NOT EQUAL THE SUM OF THE INPUT I             INPU2110
 9906 NK=2053                                                           INPU2120
 9999 WRITE (6,8888)NK,I,J,K,L,M,N                                      INPU2130
      PRINT 8888,NK,I,J,K,L,M,N                                         INPU2140
      CALL DUMP                                                         INPU2150
 2016 RETURN                                                            INPU2160
```

```
C               FORMATS                                                 INPU2170
 8004 FORMAT(7E10.5,I2)                                                 INPU2180
 80120FORMAT (I1,71HTHIS IS THE CLAM PROGRAM AND THERE IS AN ERROR.     INPU2190
     1            )                                                     INPU2200
 8048 FORMAT(1H /9H PROB NO.F9.3,12X,2HI=I2,26X,2HJ=I2)                 INPU2210
 8064 FORMAT(1H /10H X(I) I=0,12/(5F16.6))                              INPU2220
 8065 FORMAT(1H /10H Y(J) J=0,12/(5F16.6))                              INPU2230
 8066 FORMAT(1H /11H DX(I) I=1,I2/(5F16.6))                             INPU2240
 8067 FORMAT(1H /11H DY(J) J=1,I2/(5F16.6))                             INPU2250
 8092 FORMAT(1H /13H AREA(I) I=1,I2/(5F16.6))                           INPU2260
 8100 FORMAT(1H /14H (SHELL INPUT))                                     INPU2270
 8102 FORMAT(2I1,4I2,4E10.4)                                            INPU2280
 8888 FORMAT(1H+/26H1 INPUT ERROR IN STATEMENTI5,12X,12H INDICES ARE6I7)INPU2290
      END                                                               INPU2300
```

```
$IBFTC PH1     LIST,DECK,REF
      SUBROUTINE PH1                                                    PH1 0010
C                                                                       PH1 0740
C     ***** NOTE (1 MATERIAL ONLY ((X))                                 PH1 0950
C                   READ IN GEOMETRY ETC.                               PH1 0960
      NPP=7                                                             PH1 0980
      NPR=NPP-1                                                         PH1 0990
      TPIDY=PIDY*2.0                                                    PH1 1000
      ND=0                                                              PH1 1010
      NX=0                                                              PH1 1020
      NT=1                                                              PH1 1030
      NYY=1                                                             PH1 1040
C     FIRST CARD OF EACH PACKAGE.
      READ  (5,8008)IX,LX,MX,TEMP(1),TEMP(2),TEMP(3)                    PH1 1050
C     INITIALIZE THE NUMBER OF PACKAGES TO 0.
      NPKG=0                                                            PH1 1060
 2015 IF(IX-1)9901,2018,2018                                            PH1 1070
 2016 IX=I                                                              PH1 1080
      LX=L                                                              PH1 1090
      MX=M                                                              PH1 1100
C                   IF THERE ARE NO MORE PACKAGES GO COMPUTE TOTAL VALUES  PH1 1110
C     THE LAST CARD HAS A 2 PUNCH IN COL 1.
 2017 IF(IX-2)2018,7000,9902                                            PH1 1120
 2018 J=0                                                               PH1 1130
      NFKG=NPKG+1                                                       PH1 1140
C     SET PACKAGE MASS AND ENERGY TO 0.
      PE=0.0                                                            PH1 1150
      PM=0.0                                                            PH1 1160
C     ORIGIN FOR THE RADIUS VECTORS TO BE USED
C     FOR THE FIT ROUTINES(1 THRU 6).
      YC=TEMP(1)                                                        PH1 1170
      XC=TEMP(2)                                                        PH1 1180
C     S8 CONTAINS THE FIT NUMBER FOR THE
C     PACKAGE IN QUESTION.
      S8=TEMP(3)                                                        PH1 1190
      WRITE (6,8100)(NPKG,MX)                                           PH1 1200
C     NOW READ IN THE GEOMETRY AND DENSITY,
C     ENERGY AND VELOCITY CARDS.
 2020 READ  (5,8008)I,L,M,(TEMP(N),N=1,6)                               PH1 1210
      IWS=1                                                             PH1 1220
      IF(I-5)2021,2040,2022                                             PH1 1230
C     IF=, THIS IS A RHO, VELOCITY OR ENERGY CARD.
C     IF LESS, YOU HAVE READ ALL CARDS FOR THIS
C     PACKAGE IN, PLUS THE FIRST CARD FROM THE
C     NEXT PACKAGE.
 2021 IF(I-3)2060,9903,2026                                             PH1 1240
C     IF GREATER, EITHER A TRIANGLE OR PERTURBED ELLIPSE.
 2022 IF(L)9904,2030,2024                                               PH1 1250
C     A PERTURBED ELLIPSE.
 2024 IWS=7                                                             PH1 1260
```

```
      GO TO 2030                                                        PH1 1270
 2026 IWS=3                                                             PH1 1280
 2027 IF(L)9905,2030,2028                                               PH1 1290
 2028 IWS=5                                                             PH1 1300
C     A TRIANGLE.
 2030 IF(M)9906,2034,2032                                               PH1 1310
C     IF=, DELETE THIS GEOMETRY.
 2032 IWS=IWS+1                                                         PH1 1320
 2034 J=J+1                                                             PH1 1330
C     TAB STORAGE CONTAINS THE COORDINATES OF
C     GEOMETRY.
      ITAB(J)=IWS                                                       PH1 1340
      DO 2036 N=1,NPR                                                   PH1 1350
      J=J+1                                                             PH1 1360
 2036 TAB(J)=TEMP(N)                                                    PH1 1370
      GO TO 2020                                                        PH1 1380
C               ONE ONLY RHO,I,U OR V ALLOWED PER PACKAGE               PH1 1390
C     IF=, THIS IS A DENSITY CARD.
 2040 IF(L-1)9907,2046,2042                                             PH1 1400
C     IF GREATER, EITHER A VELOCITY OR ENERGY CARD.
 2042 IF(L-3)2052,2058,9908                                             PH1 1410
C     IF=, THIS IS A VELOCITY CARD, IF LESS, THIS IS A
C     ENERGY CARD.
C               DENSITY                                                 PH1 1420
 2046 DO 2048 N=1,6                                                     PH1 1430
 2048 TABR(N)=TEMP(N)                                                   PH1 1440
      GO TO 2020                                                        PH1 1450
C               ENERGY                                                  PH1 1460
 2052 DO 2054 N=1,6                                                     PH1 1470
 2054 TABI(N)=TEMP(N)                                                   PH1 1480
      GO TO 2020                                                        PH1 1490
C               VELOCITY (U AND V)                                      PH1 1500
 2058 DO 2059 N=1,6                                                     PH1 1510
 2059 TABUV(N)=TEMP(N)                                                  PH1 1520
      GO TO 2020                                                        PH1 1530
C               OUTPUT DENSITY, ENERGY, AND VELOCITY PARAMETERS         PH1 1540
C     ALL CARDS FOR THIS PACKAGE HAVE
C     BEEN READ IN.
 2060 IF(J-JTM)2070,2070,9915                                           PH1 1550
C     NO. OF PACKAGES EXCEED (72), NOTE
C     JTM SET=TO 500 IN INPUT, THUS MAX.
C     NO. OF PACKAGES = 72, UNLESS DIMENSIONS
C     ARE CHANGED.
 2070 WRITE (6,8036)(TABR(II),II=1,6)                                   PH1 1560
      WRITE (6,8038)(TABI(II),II=1,6)                                   PH1 1570
      WRITE (6,8040)(TABUV(II),II=1,6)                                  PH1 1580
C               COMPUTE BOUNDARIES OF GEOMETRIES FOR EFFICIENCY IN      PH1 1590
C               GENERATING OR DELETING PARTICLES                        PH1 1600
 3000 CALL PH2                                                          PH1 1610
C               COMPUTE I(0),I(N),J(0) AND J(N),FROM PREVIOUSLY         PH1 1620
```

```
C                 COMPUTED VALUES,FOR UPPER AND LOWER LIMITS IN
C                 THE CELL MESH SCAN
C         IXN=MINIMUM (I) OF GEOMETRY OF PACKAGE
C         IYN=MINIMUM (J) OF GEOMETRY OF PACKAGE
C         IXX=MAXIMUM (I) OF GEOMETRY OF PACKAGE
C         IYX=MAXIMUM (J) OF GEOMETRY OF PACKAGE
 3001 IXN=1
      IXX=1
      IWS=IMAX-1
 3800 IF(IWS)9929,3820,3801
 3801 DO 3808 N=1,IWS
      IF(X(N)-GXN)3802,3804,3804
 3802 IXN=IXN+1
 3804 IF(X(N)-GXX)3806,3806,3808
 3806 IXX=IXX+1
 3808 CONTINUE
      IF(IXN)3812,3812,3814
 3812 IXN=1
 3814 IF(IMAX-IXX)3816,3818,3818
 3816 IXX=IMAX
 3818 IF(IXN-IXX)3820,3820,9930
 3820 IYN=1
      IYX=1
      IWS=JMAX-1
 3821 IF(IWS)9929,3834,3822
 3822 DO 3813 N=1,IWS
 3823 IF(Y(N)-GYN)3819,3817,3817
 3819 IYN=IYN+1
 3817 IF(Y(N)-GYX)3815,3815,3813
 3815 IYX=IYX+1
 3813 CONTINUE
      IF(IYN)3824,3824,3826
 3824 IYN=1
 3826 IF(JMAX-IYX)3828,3830,3830
 3828 IYX=JMAX
 3830 IF(IYN-IYX)3834,3834,9931
 3834 WRITE (6,8044)IXN,IYN,IXX,IYX
C                 SCAN CELL MESH TO DETERMINE IF PARTICLES ARE TO BE
C                 GENERATED OR DELETED
C                 GENERATE PARTICLES
 4000 CALL PH3
C                 REARRANGE X,Y AND M FOR PARTICLES IF NECESSARY
 6011 LA=NY-NT
      IF(LX)9947,6020,6022
 6020 ND=ND+LA
      GO TO 6024
 6022 NX=NX+LA
 6024 NT=NY
      ETH=ETH+PE
C     REPLACE 606760000000 BY-18115198976
```

```
      WS=(-ABS(-18115198976))                                           PH1 2090
 6026 IF(LX)9933,6028,6030                                              PH1 2100
C     REPLACE 244663000000 BY 22125740032                               PH1 2110
 6028 WS=  ABS( 22125740032)                                            PH1 2120
 6030 WRITE (6,8501)LA,WS,PE,PM                                         PH1 2130
C                 GO READ IN NEXT PACKAGE                               PH1 2140
 6050 GO TO 2016                                                        PH1 2150
 7000 NMAX=NT                                                           PH1 2160
C     NMAX=MAX. NUMBER OF PARTICLES+1.
C     YOU HAVE PROCESSED ALL PACKAGES, ALL
C     PARTICLES, NOW GO TO THE OUTPUT.
      IF(AM(2))4051,4050,4051                                           PH1 2170
 4050 N3=NRC                                                            PH1 2180
      GO TO 4060                                                        PH1 2190
 4051 NRC=NRC+1                                                         PH1 2200
      N3=NRC                                                            PH1 2210
C     N3=NO. OF PARTICLE RECORDS OF
C     N4 WORDS.
      IF(PROB)4052,4052,4060                                            PH1 2220
 4052 WRITE (N2)(AM(I),XL(I),YL(I),IW1(I),IW2(I),I=2,NPRI)              PH1 2230
 4060 N6=NMAX-(N4-1)*(N3-1)                                             PH1 2240
      NOPR=N3                                                           PH1 2250
      REWIND N2                                                         PH1 2260
      GO TO 10000                                                       PH1 2270
C                 ERROR                                                 PH1 2280
 9901 NK=2015                                                           PH1 2290
      GO TO 9999                                                        PH1 2300
 9902 NK=2017                                                           PH1 2310
      GO TO 9999                                                        PH1 2320
 9903 NK=2021                                                           PH1 2330
      GO TO 9999                                                        PH1 2340
 9904 NK=2022                                                           PH1 2350
      GO TO 9999                                                        PH1 2360
 9905 NK=2027                                                           PH1 2370
      GO TO 9999                                                        PH1 2380
 9906 NK=2030                                                           PH1 2390
      GO TO 9999                                                        PH1 2400
 9907 NK=2040                                                           PH1 2410
      GO TO 9999                                                        PH1 2420
 9908 NK=2042                                                           PH1 2430
      GO TO 9999                                                        PH1 2440
 9915 NK=2060                                                           PH1 2450
      GO TO 9999                                                        PH1 2460
 9929 NK=3800                                                           PH1 2470
      GO TO 9999                                                        PH1 2480
 9930 NK=3818                                                           PH1 2490
      GO TO 9999                                                        PH1 2500
 9931 NK=3830                                                           PH1 2510
      GO TO 9999                                                        PH1 2520
 9933 NK=6026                                                           PH1 2530
```

```
      GO TO 9999
 9947 NK=6011
 9999 WRITE (6,8888)NK
      PRINT 8888,NK
      CALL DUMP
10000 RETURN
C                 FORMATS
 8008 FORMAT (2I1,I5,E13.5,5E10.5)
 8036 FORMAT(1H07X,8HDENSITY 9X,1P6E16.6)
 8038 FORMAT(1H07X,8HENERGY  9X,1P6E16.6)
 8040 FORMAT(1H07X,8HVELOCITY9X,1P6E16.6/1H0/)
 8044 FORMAT(1H /6H I(1)=I2,4X,5HJ(1)=I2,4X,5HI(N)=I2,4X,5HJ(N)=I2)
81000FORMAT(1H0///12H0PACKAGE NO.I3,I20,15H PARTICLES/CELL//33X,2HA114X
     1,2HA214X,2HA314X,2HA414X,2HA514X,2HA6)
85010FORMAT(1H0/I28,2H (A3,11H) PARTICLES22X,4HPE =1PE12.6,16X,4HPM =E1
     12.6)
 8888 FORMAT(23H1PH1 ERROR IN STATEMENTI5)
      END
```

```
$IBFTC PH2     LIST,DECK,REF
      SUBROUTINE PH2                                                    PH2 0010
CALCULATE THE PACKAGE GEOMETRIES                                        PH2 0020
C                                                                       PH2 0740
C                                                                       PH2 0950
C                                                                       PH2 0960
C                GENERATING OR DELETING PARTICLES                       PH2 0970
C     J=VALUE OF LAST COORDINATE READ IN.
      JT=J                                                              PH2 0980
C     INITIALIZE OUTER BOUNDARIES.
      GXN=XMAX                                                          PH2 0990
      GYN=YMAX                                                          PH2 1000
      GXX=0.0                                                           PH2 1010
      GYX=0.0                                                           PH2 1020
C     NPP=7(SET IN PH1).
      DO 3700 J=1,JT,NPP                                                PH2 1030
C     IWS STORED IN ITAB ARRAY IN PH1.
C     IF IWS=2(A TRIANGLE),IF=4(A RECTANGLE),
C     IF=6,A ELLIPSE OR CIRCLE. IF IWS=8,A
C     PERTURBED ELLIPSE. IF IWS IS LESS THAN
C     THESE VALUES, THE DEFINITION STILL HOLDS, BUT
C     NOW DELETE THIS GEOMETRY.
      KK=(ITAB(J)-1)/2                                                  PH2 1040
 3007 IF(KK)9919,3010,3008                                              PH2 1050
 3008 IF(KK-2)3100,3200,3009                                            PH2 1060
 3009 IF(KK-4)3400,9920,9920                                            PH2 1070
C                TRIANGLE                                               PH2 1080
C     VERTICES CAN BE INPUTED IN ANY ORDER,
C     X COORDINATE FIRST.
C     SEARCH FOR THE LARGEST X(WSE) AND
C     SMALLEST X(WSD).
C     FIND MAXIMUM(WSE) AND MINIMUM(WSD) X COORDINATE                   PH2 1090
 3010 IF(TAB(J+1)-TAB(J+3))3011,3012,3013                               PH2 1100
 3011 WSE=TAB(J+3)                                                      PH2 1110
      WSD=TAB(J+1)                                                      PH2 1120
      GO TO 3014                                                        PH2 1130
 3012 TAB(J+1)=TAB(J+1)*1.0000001+1.0E-8                                PH2 1140
 3013 WSE=TAB(J+1)                                                      PH2 1150
      WSD=TAB(J+3)                                                      PH2 1160
 3014 IF(TAB(J+5)-WSD)3020,3019,3016                                    PH2 1170
 3016 IF(TAB(J+5)-WSE)3024,3017,3018                                    PH2 1180
 3017 TAB(J+5)=TAB(J+5)*1.0000001+1.0E-8                                PH2 1190
 3018 WSE=TAB(J+5)                                                      PH2 1200
      GO TO 3024                                                        PH2 1210
 3019 TAB(J+5)=TAB(J+5)*0.9999999-1.0E-8                                PH2 1220
 3020 WSD=TAB(J+5)                                                      PH2 1230
C                ARRANGE VERTICES IN ASCENDING ORDER                    PH2 1240
 3024 IF(TAB(J+2)-TAB(J+4))3036,3034,3038                               PH2 1250
 3034 TAB(J+2)=TAB(J+2)*1.0000001+1.0E-8                                PH2 1260
      GO TO 3038                                                        PH2 1270
```

```
 3036 WSA=TAB(J+1)
      WSB=TAB(J+2)
      TAB(J+1)=TAB(J+3)
      TAB(J+2)=TAB(J+4)
      TAB(J+3)=WSA
      TAB(J+4)=WSB
 3038 IF(TAB(J+4)-TAB(J+6))3042,3040,3044
 3040 TAB(J+6)=TAB(J+6)*0.9999999-1.0E-8
      GO TO 3044
 3042 WSA=TAB(J+3)
      WSB=TAB(J+4)
      TAB(J+3)=TAB(J+5)
      TAB(J+4)=TAB(J+6)
      TAB(J+5)=WSA
      TAB(J+6)=WSB
      GO TO 3024
C     WSF=MINIMUM VALUE OF Y
C     WSG=MAXIMUM VALUE OF Y
 3044 WSF=TAB(J+6)
      WSG=TAB(J+2)
C                COMPUTE SLOPES
      SLA=(TAB(J+4)-TAB(J+2))/(TAB(J+3)-TAB(J+1))
      SLB=(TAB(J+6)-TAB(J+2))/(TAB(J+5)-TAB(J+1))
 3053 IF(SLA-SLB)3054,9921,3058
 3054 IF(SLA)3056,9922,3064
 3056 IF(SLB)3064,9923,3062
 3058 IF(SLA)3062,9924,3056
 3062 WSA=TAB(J+3)
      WSB=TAB(J+4)
      WSC=SLA
      TAB(J+3)=TAB(J+5)
      TAB(J+4)=TAB(J+6)
      SLA=SLB
      TAB(J+5)=WSA
      TAB(J+6)=WSB
      SLB=WSC
 3064 IF(TAB(J+3)-TAB(J+5))3066,9925,3068
 3066 ITAB(J)=ITAB(J)+2
      IWS=ITAB(J)-3
      GO TO 3069
 3068 IWS=ITAB(J)-1
 3069 KE=J+1
      KF=KE+5
C     REPLACE 272545000000 BY 25058082816
      WS=  ABS( 25058082816)
      IF(IWS)3072,3070,3072
C     REPLACE 242543000000 BY 21836333056
 3070 WS=  ABS( 21836333056)
 3072 WRITE (6,8016)WS,(TAB(N),N=KE,KF)
      WS=TAB(J+2)-SLB*TAB(J+1)
```

```
      TAB(J+1)=TAB(J+2)-SLA*TAB(J+1)
      TAB(J+6)=(TAB(J+6)-TAB(J+4))/(TAB(J+5)-TAB(J+3))
      TAB(J+5)=TAB(J+4)-TAB(J+6)*TAB(J+3)
      TAB(J+2)=SLA
      TAB(J+3)=WS
      TAB(J+4)=SLB
      GO TO 3600
C                 RECTANGLE
 3100 ITAB(J)=ITAB(J)+2
      IWS=ITAB(J)-5
C     REPLACE 272545000000 BY 25058082816
      WS=  ABS( 25058082816)
      IF(IWS)3110,3105,3110
C     REPLACE 242543000000 BY 21836333056
 3105 WS=  ABS( 21836333056)
 3110 WRITE (6,8020)WS,TAB(J+1),TAB(J+2),TAB(J+3),TAB(J+4)
      WSD=TAB(J+1)
      WSE=TAB(J+2)
      WSF=TAB(J+3)
      WSG=TAB(J+4)
      GO TO 3600
C                 ELLIPSE OR CIRCLE
 3200 IF(ABS(TAB(J+1)-TAB(J+2))-1.0E-8)3300,3300,3202
 3202 IF(TAB(J+2))9926,3300,3203
C                 ELLIPSE WITH NO PERTURBATION
 3203 ITAB(J)=ITAB(J)+2
      IWS=ITAB(J)-7
C     REPLACE 272545000000 BY 25058082816
      WS=  ABS( 25058082816)
      IF(IWS)3210,3205,3210
C     REPLACE 242543000000 BY 21836333056
 3205 WS=  ABS( 21836333056)
 3210 WRITE (6,8024)WS,TAB(J+1),TAB(J+2),TAB(J+3),TAB(J+4)
 3215 WSD=TAB(J+3)-TAB(J+1)
      WSE=TAB(J+3)+TAB(J+1)
      WSF=TAB(J+4)-TAB(J+2)
      WSG=TAB(J+4)+TAB(J+2)
      TAB(J+1)=TAB(J+1)**2
      TAB(J+2)=TAB(J+2)**2
      GO TO 3600
C                 CIRCLE
 3300 ITAB(J)=ITAB(J)+4
      IWS=ITAB(J)-9
      TAB(J+2)=TAB(J+1)
C     REPLACE 272545000000 BY 25058082816
      WS=  ABS( 25058082816)
      IF(IWS)3310,3305,3310
C     REPLACE 242543000000 BY 21836333056
 3305 WS=  ABS( 21836333056)
 3310 WRITE (6,8028)WS,TAB(J+1),TAB(J+3),TAB(J+4)
```

```
      GO TO 3215
C                ELLIPSE WITH PERTURBATION
 3400 ITAB(J)=ITAB(J)+4
      WS=1.0-(TAB(J+5)/TAB(J+1))**2
      IWSA=ITAB(J+7)
     OTAB(J+7)=(TAB(J+6)-TAB(J+4)-TAB(J+2)*SQRT(WS))/
     1          ((TAB(J+5)*(TAB(J+5)-TAB(J+1)))**2)
      IWS=ITAB(J)-11
      KE=J+1
      KF=KE+6
C     REPLACE 272545000000 BY 25058082816
      WSA=  ABS( 25058082816)
      IF(IWS)3410,3405,3410
C     REPLACE 242543000000 BY 21836333056
 3405 WSA=  ABS( 21836333056)
 3410 WRITE (6,8032)WSA,(TAB(N),N=KE,KF)
 3415 IF(WS)9927,9927,3420
 3420 IF(TAB(J+3))9928,3425,9928
 3425 TAB(J+3)=TAB(J+7)
      ITAB(J+7)=IWSA
      WSA=TAB(J+2)+TAB(J+2)/4.0
      WSD=0.0
      WSE=TAB(J+1)+TAB(J+1)/4.0
      WSF=TAB(J+4)-WSA
      WSG=TAB(J+4)+WSA
C                DETERMINE BOUNDARIES OF GEOMETRIES
 3600 IF(WSD-GXN)3602,3604,3604
C     MAXIMUM (X)
 3602 GXN=WSD
 3604 IF(WSE-GXX)3608,3608,3606
C     MINIMUM (X)
 3606 GXX=WSE
 3608 IF(WSF-GYN)3610,3612,3612
C     MAXIMUM (Y)
 3610 GYN=WSF
 3612 IF(WSG-GYX)3700,3700,3614
C     MINIMUM (Y)
 3614 GYX=WSG
 3700 CONTINUE
      J=JT
      GO TO 10000
C                E R R O R
 9919 NK=3007
      GO TO 9999
 9920 NK=3009
      GO TO 9999
 9921 NK=3053
      GO TO 9999
 9922 NK=3054
      GO TO 9999
```

```
 9923 NK=3056
      GO TO 9999
 9924 NK=3058
      GO TO 9999
 9925 NK=3064
      GO TO 9999
 9926 NK=3202
      GO TO 9999
 9927 NK=3415
      GO TO 9999
 9928 NK=3420
 9999 WRITE (6,8888)NK
      PRINT 8888,NK
      CALL DUMP
10000 RETURN
 8016 FORMAT(15HOTRIANGLE ---- A3,7H ------1P6E16.6)
 8020 FORMAT(15HORECTANGLE --- A3,7H ------1P6E16.6)
 8024 FORMAT(15HOELLIPSE ----- A3,7H ------1P6E16.6)
 8028 FORMAT(15HOCIRCLE ------ A3,7H ------1PE16.6,16X,4E16.6)
 8032 FORMAT(15HOP ELLIPSE --- A3,7H ------1P6E16.6)
 8888 FORMAT(23H1PH2 ERROR IN STATEMENTI5)
      END
```

```
$IBFTC PH3     LIST,DECK,REF
      SUBROUTINE PH3                                                    PH3 0010
C     GENERATE (OR DELETE) THE PARTICLES                                PH3 0020
C                                                                       PH3 0740
C                                                                       PH3 0950
C                                                                       PH3 0960
C                                                                       PH3 0970
C                SCAN CELL MESH TO DETERMINE IF PARTICLES ARE TO BE     PH3 0980
C                GENERATED OR DELETED                                   PH3 0990
C                G E N E R A T E   P A R T I C L E S                    PH3 1000
C     SAVE CURRENT VALUES OF COUNTERS.
 4000 IA=I                                                              PH3 1010
      JA=J                                                              PH3 1020
      IJ=K                                                              PH3 1030
      JT=L                                                              PH3 1040
      IF(IX-1)9932,4010,9932                                            PH3 1050
 4010 IF(MX-MNP)4012,4012,9935                                          PH3 1060
C     IF GREATER, YOU TRIED TO GENERATE MORE THAN
C     400 PARTICLES / CELL.
 4012 WS=MX                                                             PH3 1070
      FMX=SQRT(WS)                                                      PH3 1080
      MXS=FMX+.5                                                        PH3 1090
 4011 IF(MXS*MXS-MX)9936,4013,9936                                      PH3 1100
C     IF(GREATER OR LESS) THE NO. OF PARTICLES / CELL
C     THAT YOU REQUESTED WAS NOT N SQ. WHERE
C     N IS FROM 1 TO 20.
 4013 MXA=1-MX                                                          PH3 1110
      TFMX=.5/FMX                                                       PH3 1120
      WPIDY=TPIDY/FMX                                                   PH3 1130
 4015 IF(MXA)4018,4018,9937                                             PH3 1140
C     IF GREATER, YOU HAVE FAILED TO SPECIFY THE
C     NO. OF PARTICLES TO GENERATE.
 4018 NY=NT                                                             PH3 1150
      DO 5700 I=IXN,IXX                                                 PH3 1160
C                COMPUTE THE COORDINATE OF THE PARTICLE                 PH3 1170
C                UNDER CONSIDERATION                                    PH3 1180
      WS5=DX(I)/FMX                                                     PH3 1190
C     THE VOLUME OF THE SUBDIVIDED CELL =
C     PI(2.*XL(N)DY/N*DY/N).
      TABX(1)=X(I)-TFMX*DX(I)                                           PH3 1200
 4019 IF(MXA)4020,4024,9938                                             PH3 1210
 4020 DO 4022 K=2,MXS                                                   PH3 1220
C     WE START AT THE RIGHT AND TOP OF CELL(K).
C     SET UP ARRAY FOR X COORDINATES OF THE
C     PARTICLES.
 4022 TABX(K)=TABX(K-1)-WS5                                             PH3 1230
C     J LOOP, LIMITS OF Y FOR THIS PACKAGE.
 4024 DO 5700 J=IYN,IYX                                                 PH3 1240
      TAM=WPIDY*WS5*DY(J)                                               PH3 1250
C     TAM= 2PI/N*DX/N*DY
```

```
      E=0.0                                                             PH3 1260
      IIWS=0                                                            PH3 1270
      IWS=0                                                             PH3 1280
      IB=0                                                              PH3 1290
      WS=DY(J)/FMX                                                      PH3 1300
      TABY(1)=Y(J)-TFMX*DY(J)                                           PH3 1310
C     MXS=N
      DO 4026 K=2,MXS                                                   PH3 1320
C     SET UP ARRAY FOR Y COORDINATES OF THE
C     PARTICLES.
 4026 TABY(K)=TABY(K-1)-WS                                              PH3 1330
C     K USED FOR THE CELL QUANTITIES.
      K=(J-1)*IMAX+I+1                                                  PH3 1340
 4028 IBB=IB/MXS                                                        PH3 1350
      IB=IB+1                                                           PH3 1360
      IBA=MOD(IB,MXS)                                                   PH3 1370
C     TX=X COORDINATE OF PARTICLE IN QUESTION.
      TX=TABX(IBA+1)                                                    PH3 1380
C     TY=Y COORDINATE OF PARTICLE IN QUESTION.
      TY=TABY(IBB+1)                                                    PH3 1390
C                GENERATE OR DELETE THE PARTICLE                        PH3 1400
      ID=0                                                              PH3 1410
      IG=0                                                              PH3 1420
      DO 4200 L=1,JA,NPP                                                PH3 1430
      KK=ITAB(L)                                                        PH3 1440
      IF(KK-5)4062,4078,4078                                            PH3 1450
C                TRIANGLE                                               PH3 1460
 4062 WSX=(TY-TAB(L+1))/TAB(L+2)                                        PH3 1470
      IF(WSX-TX)4064,4064,4200                                          PH3 1480
 4064 WSX=(TY-TAB(L+3))/TAB(L+4)                                        PH3 1490
      IF(WSX-TX)4200,4066,4066                                          PH3 1500
 4066 WSY=TAB(L+6)*TX+TAB(L+5)                                          PH3 1510
      IF(KK-2)4068,4068,4072                                            PH3 1520
 4068 IF(WSY-TY)4200,4070,4070                                          PH3 1530
 4070 GO TO (4074,4076,4074,4076),KK                                    PH3 1540
 4072 IF(WSY-TY)4070,4070,4200                                          PH3 1550
 4074 ID=1                                                              PH3 1560
      GO TO 4200                                                        PH3 1570
 4076 IG=1                                                              PH3 1580
      GO TO 4200                                                        PH3 1590
 4078 KK=KK-4                                                           PH3 1600
 4077 IF(KK-8)4079,4094,9939                                            PH3 1610
 4079 GO TO (4080,4080,4090,4090,4092,4092,4094),KK                     PH3 1620
C                RECTANGLE                                              PH3 1630
 4080 IF(TAB(L+1)-TX)4082,4082,4200                                     PH3 1640
 4082 IF(TAB(L+2)-TX)4200,4084,4084                                     PH3 1650
 4084 IF(TAB(L+3)-TY)4086,4086,4200                                     PH3 1660
 4086 IF(TAB(L+4)-TY)4200,4088,4088                                     PH3 1670
 4088 GO TO (4074,4076),KK                                              PH3 1680
C                ELLIPSE WITH NO PERTURBATION                           PH3 1690
```

```
 4090 KK=KK-2
      IF((TX-TAB(L+3))**2/TAB(L+1)+(TY-TAB(L+4))**2
     1/TAB(L+2)-1.0)4088,4088,4200
C                CIRCLE
 4092 KK=KK-4
     0IF((TX-TAB(L+3))**2+(TY-TAB(L+4))**2-TAB(L+1))
     1  4088,4088,4200
C                ELLIPSE WITH PERTURBATION
 4094 KK=KK-6
     0IF((TX/TAB(L+1))**2+(TY-TAB(L+4)-TAB(L+3)*(TX*
     1  (TX-TAB(L+1))))**2)**2/TAB(L+2)-1.0)4088,4088,4200
 4200 CONTINUE
C                IF ID=1 DELETE
 4201 IF(ID)9940,4310,4800
C                IF ID=0 AND IG=0 DELETE
 4310 IF(IG)9941,4800,4312
C                GENERATE PARTICLE
 4312 NY=NY+1
      IF(IIWS)23,22,23
   22 IIWS=1
   23 IWS=1
      NYY=NYY+1
      CALL PH4
C     RETURN FROM PH4 WITH THE FOLLOWING DATA,
C     WSR=PARTICLE DENSITY
C     WSI=PARTICLE SPECIFIC INTERNAL ENERGY
C     WSU=RADIAL VELOCITY COMPONENT OF PARTICLE
C     WSV=AXIAL VELOCITY COMPONENT OF PARTICLE
 4332 N=NYY
      IF(IIWS)4335,4335,24
   24 IIWS=-1
 4333 IF(AMX(K))9951,4335,4334
C     CALCULATE PACKAGE ENERGY.
 4334 E=((U(K)**2+V(K)**2)/(AMX(K)))*.5+AIX(K)
C     SET THE PARTICLE COORDINATES INTO THE
C     PROPER ARRAYS.
 4335 XL(N)=TX
      YL(N)=TY
C     SET I AND J OF CELL K(LOCATION OF PARTICLE).
      IW1(N)=I
      IW2(N)=J
C     CALCULATE PARTICLE MASS AS
C     =2PI/N*DX/N*DY*XL(N)*RHO.
      AM(N)=TAM*TX*WSR
 4341 IF(LX)9945,4342,4344
 4342 WS=AM(N)*WSI
      IF(AM(N)-AMDM)16,15,15
   16 AMDM=AM(N)
   15 CONTINUE
      PM=PM+AM(N)
```

```
      AM(N)=-AM(N)
      GO TO 4346
 4344 WS=AM(N)*WSI
      IF(AM(N)-AMXM)18,17,17
   18 AMXM=AM(N)
   17 AIX(K)=AIX(K)+WS
      PM=PM+AM(N)
C     SUM UP MASS, BOTH COMPONENTS OF MOMENTA
C     AND TOTAL INTERNAL ENERGY IN CELL K.
      AMX(K)=AMX(K)+AM(N)
 4346 U(K)=U(K)+ABS(AM(N))*WSU
      V(K)=V(K)+ABS(AM(N))*WSV
      IF(NY-NPRR)4800,14,9945
   14 NRC=NRC+1
      NPRR=NPRR+NPRI-1
      IF(PROB)5000,5000,5001
C     WRITE PARTICLES ON TAPE IF THIS IS TO
C     BE A PIC RUN.
 5000 WRITE (N2)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,NPRI)
 5001 NYY=1
    3 DO 2 N=2,NPRI
C     SET PARTICLE ARRAYS TO ZERO.
      XL(N)=0.0
      YL(N)=0.0
      AM(N)=0.0
      IW1(N)=0
      IW2(N)=0
    2 CONTINUE
 4800 IF(MX-2B)9946,4880,4028
C A L C U L A T E    E N E R G Y    F O R    P K G
 4880 IF(IWS)4900,5700,4900
 4900 IF(AMX(K))9951,5700,4910
 4910 PEE=(U(K)**2+V(K)**2)/(AMX(K))*.5+AIX(K)
 4930 IF(E)4950,4940,4940
 4940 PEE=PEE-E
 4950 PE=PE+PEE
 5700 CONTINUE
      I=IA
      J=JA
      K=IJ
      L=JT
      GO TO 10000
C                E R R O R
 9932 NK=4000
      GO TO 9999
 9935 NK=4010
      GO TO 9999
 9936 NK=4011
      GO TO 9999
 9937 NK=4015
```

```
      GO TO 9999
 9938 NK=4019
      GO TO 9999
 9939 NK=4077
      GO TO 9999
 9940 NK=4201
      GO TO 9999
 9941 NK=4310
      GO TO 9999
 9945 NK=4341
      GO TO 9999
 9946 NK=4800
      GO TO 9999
 9951 NK=4905
 9999 WRITE (6,8888)NK,I,J,K,L,M,N
      PRINT 8888,NK,I,J,K,L,M,N
      CALL DUMP
10000 RETURN
 8888 FORMAT(1H+/26H1 P H 3 ERROR IN STATEMENTI5,12X,12H INDICES ARE6I7)
      END
```

```
$IBFTC PH4     LIST,DECK,REF
      SUBROUTINE PH4
C
C
C     THE ACTUAL COORDINATES USED IN THE FIT
C     SUBROUTINES IS TTX=TX-XC,TTY=TY-YC.
      TTX=TX-XC
      TTY=TY-YC
      LL=S8
      GO TO(1,2,3,4,5,6),LL
    1 CALL FIT1
      GO TO 7
    2 CALL FIT2
      GO TO 7
    3 CALL FIT3
      GO TO 7
    4 CALL FIT4
      GO TO 7
    5 CALL FIT5
      GO TO 7
    6 CALL FIT6
    7 RETURN
      END
```

SUBROUTINE PH4 DETERMINES WHICH ONE OF THE SIX POSSIBLE FITS TO CALL FOR, TO ASSIGN A DENSITY ENERGY, AND VELOCITIES TO PARTICLE N. THE FIT NUMBER IS SPECIFIED IN COLUMNS 31'40 OF THE FIRST CARD OF EACH PACKAGE(SEE SECTION 2.1)

```
$IBFTC FIT1    LIST,DECK,REF
      SUBROUTINE FIT1                                                   FIT10010
C                                                                       FIT10730
C                                                                       FIT10940
      WS=SQRT(TTX**2+TTY**2)                                            FIT10950
C     DENSITY                                                           FIT10960
      WSR=TABR(1)+TABR(2)*(TTY-TABR(3))                                 FIT10970
C     ENERGY                                                            FIT10980
      WSI=TABI(1)+TABI(2)*(TTY-TABI(3))                                 FIT10990
C     VELOCITIES                                                        FIT11000
      WS=TABUV(1)+TABUV(2)* TTY-TABUV(3))                               FIT11010
      WSU=0.0                                                           FIT11020
      WSV=WS                                                            FIT11030
      RETURN                                                            FIT11040
      END                                                               FIT11050
```

```
$IBFTC FIT2    LIST,DECK,REF
      SUBROUTINE FIT2                                                   FIT20010
C                                                                       FIT20730
C                                                                       FIT20940
      WS=SQRT(TTX**2+TTY**2)                                            FIT20950
C     DENSITY                                                           FIT20960
      WSR=((TTX-TABR(1))/TABR(2))**2+((TTY-TABR(3))/                    FIT20970
     1TABR(4))**2                                                       FIT20980
C     ENERGY                                                            FIT20990
      WSI=TABI(1)+TABI(2)*TTX+TABI(3)*TTX**2                            FIT21000
     1+TABI(4)*TTY+TABI(5)*TTY**2                                       FIT21010
C     VELOCITIES                                                        FIT21020
      WSV=TABUV(1)+TABUV(2)*TTY                                         FIT21030
      WSU=TABUV(3)+TABUV(4)*TTY                                         FIT21040
      RETURN                                                            FIT21050
      END                                                               FIT21060
```

```
$IBFTC FIT3    LIST,DECK,REF
      SUBROUTINE FIT3                                                   FIT30010
C                                                                       FIT30730
C                                                                       FIT30940
C      THIS FIT FOR SIN KZ/KZ ************
      WS=SQRT(TTX**2+TTY**2)                                            FIT30950
C     DENSITY                                                           FIT30960
      WSR=TABR(1)+TABR(2)*(TTY-TABR(3))
      WSA=TTY/TABI(2)
      WSB=WSA*PIDY*2.
      WSC=SIN(WSB)
      WSI=WSC/WSA*TABI(1)
      WS=TABUV(1)+TABUV(2)*(TTY-TABUV(3))
      WSU=0.
      WSV=WS
      WSI=WSI*TABI(3)
C     TABI(3) US SCALE FACTOR FOR YIELD NORMALLY SET TO 1.
      RETURN
      END



$IBFTC FIT4    LIST,DECK,REF
      SUBROUTINE FIT4                                                   FIT40010
      RETURN                                                            FIT40020
      END                                                               FIT40030


$IBFTC FIT5    LIST,DECK,REF
      SUBROUTINE FIT5                                                   FIT50010
      RETURN                                                            FIT50020
      END                                                               FIT50030


$IBFTC FIT6    LIST,DECK,REF
      SUBROUTINE FIT6                                                   FIT60010
      RETURN                                                            FIT60020
      END                                                               FIT60030
```

```
$IBFTC OUTPUT  LIST,DECK,REF
      SUBROUTINE OUTPUT                                                 OUTP0010
C        C   O   M   M   O   N                                          INPU0720
C L A M                ******   O U T P U T   ******                    OUTP0020
C                                                                       OUTP0030
C                                                                       OUTP0750
C     NOTE (1 MATERIAL ONLY ((X))                                       OUTP0960
C                                                                       OUTP0970
C                 PACKAGES HAVE BEEN READ IN AND PROCESSED              OUTP0980
C                 COMPUTE TOTAL ENERGIES AND TOTAL MASSES               OUTP0990
      E=ETH                                                             OUTP1000
      WRITE (6,8104)                                                    OUTP1010
 7001 ND=ND+1                                                           OUTP1020
      IF(E)6000,6000,6001                                               OUTP1030
 6000 AMDM=0.0                                                          OUTP1040
      AMXM=0.0                                                          OUTP1050
      GO TO 7016                                                        OUTP1060
 6001 AMDM=AMDM/2.0                                                     OUTP1070
      AMXM=AMXM/2.0                                                     OUTP1080
 7013 IF(AMDM)9901,9901,7014                                            OUTP1090
 7014 IF(AMXM)9902,9902,7016                                            OUTP1100
 7016 ETH=0.0                                                           OUTP1110
      TMDZ=0.0                                                          OUTP1120
      TMXZ=0.0                                                          OUTP1130
      DO 7012 I=2,KMAX                                                  OUTP1140
 7005 IF(AMX(I))9904,7012,7006                                          OUTP1150
 7006 CONTINUE                                                          OUTP1160
C     SUM UP TOTAL (X) MASS IN GRID.
      TMXZ=AMX(I)+TMXZ                                                  OUTP1170
C     CALCULATE SPECIFIC INTERNAL ENERGY/CELL (K).
      AIX(I)=AIX(I)/AMX(I)                                              OUTP1180
 7008 WS=AMX(I)                                                         OUTP1190
C     CALCULATE RADIAL AND AXIAL VELOCITIES BY
C     CONSERVING BOTH COMPONENTS OF MOMENTA.
      U(I)=U(I)/WS                                                      OUTP1200
      V(I)=V(I)/WS                                                      OUTP1210
C     SUM UP TOTAL ENERGY IN SYSTEM.
      ETH=ETH+((U(I)**2+V(I)**2)/2.0+AIX(I))*WS                         OUTP1220
      GO TO 7012                                                        OUTP1230
 7012 CONTINUE                                                          OUTP1240
      TMZ=TMDZ+TMXZ                                                     OUTP1250
      WRITE (6,8072)ETH,E,TMDZ,TMXZ,TMZ                                 OUTP1260
      IWS=ND-1                                                          OUTP1270
      IWSA=NMAX-ND                                                      OUTP1280
      IWSB=NMAX-1                                                       OUTP1290
      WRITE (6,8073)(IWS,IWSA,IWSB)                                     OUTP1300
C                 PUT INPUT ON BINARY TAPE 7                            OUTP1310
 7113 REWIND N7                                                         OUTP1320
C                                                                       OUTP1330
C     WRITE TAPE FOR OIL CODE.
```

```
C
      IF(PROB)7162,7162,7163
 7163 N3=0
 7162 WS=555.0
      WRITE (N7)WS,CYCLE,N3
      WRITE (N7)(Z(I),I=1,MZ)
 7131 WRITE (N7)(U(K),V(K),AMX(K),AIX(K),AIX(K),K=1,KMAXA)
      GO TO 7140
 7140 CONTINUE
      WRITE (N7)X(0),(X(K),TAU(K),K=1,IMAX)
      WRITE (N7)(Y(K),K=0,JMAX)
      WS=666.0
C
C     WRITE PARTICLES ON DUMP TAPE FOR PIC RUN.
      IF(PROB)7150,7150,7161
 7150 DO 7160 I=1,N3
      READ  (N2)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,NPRI)
      WRITE (N7)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,NPRI)
 7160 CONTINUE
 7161 WRITE (N7)WS,WS,WS
      REWIND N7
      WRITE (6,8120)T,NC
      IWS=IMAX*JMAX+1
      CALL SLITE (0)
      DO 7517 I=1,IMAX
      CALL SLITE (1)
      J=JMAXA
      K=IWS+I
      DO 7517 JP=1,JMAX
      J=J-1
      K=K-IMAX
 7170 IF(AMX(K))9905,7517,7175
 7175 CALL SLITET(1,K000FX)
       GO TO(7180,7185),K000FX
C     PRINT OUT CELL QUANTITIES.
 7180 WRITE (6,8080)I,X(I),DX(I)
 71850WRITE (6,8084)J,Y(J),DY(J),U(K),V(K),                           U(K),AIX(K)
     1,V(K),AMX(K)
 7517 CONTINUE
      IF(Q000FL)7520,7520,7616
 7616 REWIND N2
      GO TO 7520
C                ERROR
 9901 NK=7013
      GO TO 9999
 9902 NK=7014
      GO TO 9999
 9904 NK=7005
      GO TO 9999
 9905 NK=7170
```

```
 9999 WRITE (6,8888)NK,I,J,K,L,M,N                                      OUTP1910
      PRINT 8888,NK,I,J,K,L,M,N                                         OUTP1920
      CALL DUMP                                                         OUTP1930
 7520 RETURN                                                            OUTP1940
C                FORMATS                                                OUTP1950
 80720FORMAT(1H ////6H THE =1PE16.9,7X,3HE =E16.9///5H M. =E11.5,5X,4HMXOUTP1960
     1 =E11.5,7X,7HM.+MX =E11.5)                                        OUTP1970
 8073 FORMAT(1H0/17H0PARTICLES  - - -I12,4H DOTI14,2H XI14,6H TOTAL)    OUTP1980
 80800FORMAT(1H0///3H0I=I2,10X,2HX=1PE13.7,10X,3HDX=E13.7/3H0 J10X,1HY13OUTP1990
     1X,2HDY12X,1HU13X,1HV12X,3HAID11X,3HAIX11X,3HAMD11X,3HAMX)         OUTP2000
 8084 FORMAT(I3,3X,1P8E14.7)                                            OUTP2010
 8104 FORMAT(1H /31H THERE ARE NO MORE PACKAGES----)                    OUTP2060
 8120 FORMAT(1H ///18H TAPE DUMP AT TIMEF10.1,7X,5HCYCLEI4)             OUTP2070
 8888 FORMAT(1H+/26H1OUTPUT ERROR IN STATEMENTI5,12X,12H INDICES ARE6I7)OUTP2080
      END                                                               OUTP2090
```

```
C
C      *** NOTE,THE FOLLOWING SET OF DIMENSIONS,COMMON
C      AND EQUIVALENCE CARDS ARE TO BE USED FOR ALL
C      SUBROUTINES WITH THE EXCEPTION OF MAIN AND CARDS.
C
C
C
C                 D    I    M    E    N    S    I    O    N                          PH2 0020
C                                                                                    PH2 0030
       DIMENSION  AM(130),    XL(130),    YL(130),                                   PH2 0040
      1U(3500),V(3500),AMX(3500),AIX(3500),                                          PH2 0050
      2P(3500),                                                                      PH2 0060
      3 IW1(130), IW2(130),                                                          PH2 0070
      4DX(52),    X(53),      XX(54),     DY(100),    Y(100),     YY(101),           PH2 0080
      5TAB(15),   AMK(15),    PK(15),     QK(15),     Z(150),     IZ(150),           PH2 0090
      6TAU(52),   PL(200),    PR(200),    UL(200),    UR(200),                       PH2 0100
      7FLEFT(100),YAMC(100),  SIGC(100),  GAMC(100)                                  PH2 0110
       COMMON           Z        ,XX      ,UR      ,PR      ,YY                      PH2 0120
       COMMON           AID      ,AIX     ,AM      ,AMD     ,AMX     ,AREA           PH2 0130
       COMMON           BIG      ,BOUNCE  ,DDXN    ,DDVK    ,DKE     ,DVK            PH2 0140
       COMMON           DX       ,DY      ,E       ,FD      ,FS      ,FX             PH2 0150
       COMMON           OUT      ,P       ,PABOVE  ,PBLO    ,PIDTS   ,PPABOV         PH2 0160
       COMMON           PRR      ,PUL     ,QDT     ,RC      ,REZ     ,RHO            PH2 0170
       COMMON           RL,RR,SIG,Q000FL,SWITCH ,TABLM                               PH2 0180
       COMMON           TAU      ,TAUDTS  ,TAUDTX  ,U       ,UK      ,URR            PH2 0190
       COMMON           UT       ,UU      ,UUU     ,UTEF    ,UVMAX   ,V              PH2 0200
       COMMON           VABOVE   ,VBLO    ,VEL     ,VK      ,VT      ,VTEF           PH2 0210
       COMMON           VV       ,VVABOV  ,VVBLO   ,W2      ,W3      ,WPS            PH2 0220
       COMMON           WS       ,WSA     ,WSB     ,WSC     ,XL      ,XLF            PH2 0230
       COMMON           XN       ,XR      ,YL      ,YLW     ,YN      ,YU             PH2 0240
       COMMON           ZMAX     ,I       ,II      ,IN      ,IR      ,IWS            PH2 0250
       COMMON           IWSA     ,IWSB    ,IWSC    ,IW1     ,J       ,JN             PH2 0260
       COMMON           JP       ,JR      ,K       ,KN      ,KP      ,KR             PH2 0270
       COMMON           KRM      ,L       ,M       ,MA      ,MB      ,MC             PH2 0280
       COMMON           MD       ,ME      ,MZ      ,N       ,NK      ,NKMAX          PH2 0290
       COMMON           NK1      ,NO      ,NR      ,IW2                              PH2 0300
C                                                                                    PH2 0390
C                                                                                    PH2 0400
C                 E    Q    U    I    V    A    L    E    N    C    E               PH2 0410
C                                                                                    PH2 0420
      QEQUIVALENCE       (Z,IZ,PROB),        (Z(2),CYCLE),        (Z(3),DT),         PH2 0430
      1(Z(4),PRINTS),    (Z(5),PRINTL),      (Z(6),DUMPT7),       (Z(7),CSTOP),      PH2 0440
      2(Z(8),PIDY),      (Z(9),TMZ),         (Z(10),GAM),         (Z(11),GAMD),      PH2 0450
      3(Z(12),GAMX),     (Z(13),ETH),        (Z(14),FFA),         (Z(15),FFB),       PH2 0460
      4(Z(16),TMDZ),     (Z(17),TMXZ),       (Z(18),XMAX),        (Z(19),TXMAX),     PH2 0470
      5(Z(20),TYMAX),    (Z(21),AMDM),       (Z(22),AMXM),        (Z(23),DNN),       PH2 0480
      6(Z(24),DMIN),     (Z(25),FEF),        (Z(26),DTNA),        (Z(27),CVIS),      PH2 0490
      7(Z(28),NPR),      (Z(29),NPRI),       (Z(30),NC),          (Z(31),NPC),       PH2 0500
      8(Z(32),NRC),      (Z(33),IMAX),       (Z(34),IMAXA),       (Z(35),JMAX),      PH2 0510
      9(Z(36),JMAXA),    (Z(37),KMAX),       (Z(38),KMAXA),       (Z(39),NMAX)       PH2 0520
```

```
      OEQUIVALENCE       (Z(40),ND),         (Z(41),KDT),        (Z(42),IXMAX),
      1(Z(43),NOJ),      (Z(44),NOPR),       (Z(45),NIMAX),      (Z(46),NJMAX),
      2(Z(47),I1),       (Z(48),I2),         (Z(49),I3),         (Z(50),I4),
      3(Z(51),N1),       (Z(52),N2),         (Z(53),N3),         (Z(54),N4),
      4(Z(55),N5),       (Z(56),N6),         (Z(57),N7),         (Z(58),N8),
      5(Z(59),N9),       (Z(60),N10),        (Z(61),N11),        (Z(62),NRM),
      6(Z(63),TRAD),     (Z(64),XNRG),       (Z(65),SN),         (Z(66),DXN),
      7(Z(67),RADER),    (Z(68),RADET),      (Z(69),RADEB),      (Z(70),DTRAD),
      8(Z(71),REZFCT),   (Z(72),RSTOP),      (Z(73),SHELL),      (Z(74),BBOUND),
      9(Z(75),TOZONE),   (Z(76),ECK),        (Z(77),SBOUND),     (Z(78),X1)
      OEQUIVALENCE       (Z(79),X2),         (Z(80),Y1),         (Z(81),Y2),
      1(Z(82),CABLN),    (Z(83),VISC),       (Z(84),T),          (Z(85),GMAX),
      2(Z(86),WSGD),     (Z(87),WSGX),       (Z(88),GMADR),      (Z(89),GMAXR),
      3(Z(90),S1),       (Z(91),S2),         (Z(92),S3),         (Z(93),S4),
      4(Z(94),S5),       (Z(95),S6),         (Z(96),S7),         (Z(97),S8),
      5(Z(98),S9),       (Z(99),S10)
C
      OEQUIVALENCE       (XX(2),X(1)),       (UR,UL,FLEFT),      (UR(100),YAMC),
      1(PR(100),SIGC),(PR,PL,GAMC),(UR,TAB),
      2(UR(16),AMK),     (UR(31),PK),        (UR(46),QK),        (YY(2),Y(1))
C
C
C
C
C     NOTE, THERE ARE 2 SPECIAL SUBROUTINES(FORTRAN 4) USED
C     IN THE OIL CODE, SUBROUTINE SLITE SERVES THE SAME FUNCTION
C     AS TURNING ON SENSE LIGHTS, AND SLITET SERVES THE
C     FUNCTION OF TESTING THE SENSE LIGHTS.
C
C
C     NOTE, IF AN ERROR(SEE THE END OF THE SUBROUTINES) OCCURS, THE
C     SUBROUTINE WILL CALL FOR A DUMP. BY CHECKING THE
C     VAUES OF NR AND NK , ONE CAN READILY IDENTIFY THE
C     STATEMENT NUMBER AND THE SUBROUTINE WHERE THE ERROR OCCURRED.
C
C     NK WILL CONTAIN THE STSTEMENT NUMBER, AND NR IS
C     AN IDENTIFATION FOR THE SUBROUTINE AS FOLLOWS,
C     NR=1   INPUT
C     NR=2    CDT
C     NR=3    PH1
C     NR=4    PH2
C     NR=6   EDIT
```

```
$IBFTC MAIN    LIST,DECK,REF
CMAIN                                                                   MAIN0010
C                                                                       MAIN0020
C      ****** NOTE 1 MATERIAL ONLY (XI) ************                    MAIN0030
C                                                                       MAIN0050
C      INPUT READS OIL DUMP TAPE OR
C      WILL CALL SUBROUTINE SET*UP WHICH
C      WILL MAKE A DUMP TAPE FOR CERTAIN TYPES OF PROBLEM
C      (SEE SECTION ON SET*UP)
C      ALSO CALCULATES DX AND DY AND EQUATION OF STATE DATA .....
       CALL INPUT                                                       MAIN0060
C      CDT ROUTINE CALCULATES DT(HYDRO TIME STEP)
C      AND PRESSURES, ADVANCE CYCLE NO. ETC.
   10  CALL CDT                                                         MAIN0070
C      IN EDIT, DETERMINE WHETHER TO EXECUTE A LONG
C      PRINT, A SHORT PRINT, A TAPE DUMP, ETC. AND
C      CALCULATE TOTAL ENERGY IN SYSTEM(COMPARE
C      WITH ETH) TOTAL MASS, INTEGRATE TOTAL
C      COMPONENTS OF MOMENTA.
       CALL EDIT                                                        MAIN0080
       CALL SLITET(1,K000FX)                                            MAIN0090
C       SENSE LITE 1 SIGNIFIES THIS
C      IS THE LAST CYCLE OF THIS RUN $$$$$$$$$$$$$$$
C      LITE TURNED ON IN THE EDIT ROUTINE ******
        GO TO(30,20),K000FX                                             MAIN0100
C      PH1, INTEGRATE THE MOMENTA EQS. INTEGRATE
C      ENERGY EQUATION(ONLY CHANGES DUE TO WORK
C       TERMS). NO MOVEMENT OF MASS HERE
   20  CALL PH1                                                         MAIN0110
C      TRANSPORT MASS ACROSS BOUNDARIES (SOLVE
C      MASS TRANSPORT EQ.) TRANSPORT TERMS IN
C      THE MOMENTA AND ENERGY EQS. LEFT OUT OF
C      PH1, HERE APROXIMATED BY MASS MOVEMENT. CONSERVE
C      MASS, MOMENTA AND TOTAL ENERGY.
       CALL PH2                                                         MAIN0120
C                                                                       MAIN0130
C                                                                       MAIN0140
       GO TO 10                                                         MAIN0150
   30  CALL EXIT                                                        MAIN0160
       END                                                              MAIN0170
```

```
$IBFTC INPUT   LIST,DECK,REF
      SUBROUTINE INPUT                                                  INPU0010
C                                                                       INPU0760
C                                                                       INPU0900
C     TURN ON  SENSE  LITE  3.
      CALL SLITE (3)                                                    INPU0980
C                                                                       INPU0990
C     READ HEADER CARD (COLUMNS 2-72).
      READ  (5,8004)IWS                                                 INPU1000
      WRITE (6,8004)IWS                                                 INPU1010
C     CALL DATA.
    6 CALL CARDS                                                        INPU1020
C     IF PK(3) = OR GREATER THAN ZERO, CALL ROUTINE
C     SET-UP, OTHERWISE, BINARY OIL TAPE HAS BEEN MADE.
C     READ IN DATA FROM OIL DUMP TAPE, OR
C     GENERATE A DUMP TAPE FOR OIL, AND
C     CALCULATE DX AND DY FROM THE X AND
C     Y VALUES FROM TAPE.
      IF(PK(3))8687,8888,8888                                           INPU1030
 8888 CALL CARDS                                                        INPU1040
      CALL SETUP                                                        INPU1050
 8887 CONTINUE                                                          INPU1060
C                                                                       INPU1070
C               READ TAPE                                               INPU1080
C     GO READ BINARY TAPE.
      GO TO 1000                                                        INPU1090
C                                                                       INPU1100
C               READ IN REMAINING INPUT CARDS                           INPU1110
   10 CONTINUE                                                          INPU1120
      CALL CARDS                                                        INPU1130
      GO TO 2000                                                        INPU1140
C                                                                       INPU1150
C     SET THE PRESSURES TO ZERO.
   40 DO 45 K=1,KMAXA                                                   INPU1160
   45 P(K)=0.0                                                          INPU1170
C     INTEGRATE BACKWARDS ON CYCLE, TIME AND NO. OF
C     CYCLES BETWEEN ENERGY CHECK, SINCE THESE
C     ARE ADVANCED IN CDT.
      T=T-DTNA                                                          INPU1180
      NC=NC-1                                                           INPU1190
      CYCLE=NC                                                          INPU1200
      NPC=NPC-1                                                         INPU1210
      UVMAX=0.0                                                         INPU1220
C     CALCULATE THE DX'S, SINCE THESE ARE NOT ON
C     TAPE.
      DO 50 I=1,IMAX                                                    INPU1230
   50 DX(I)=X(I)-X(I-1)                                                 INPU1240
C     CALCULATE THE DY'S, SINCE THESE ARE NOT ON
C     TAPE.
      DO 55 J=1,JMAX                                                    INPU1250
```

```
   55 DY(J)=Y(J)-Y(J-1)                                                  INPU1260
      J=MZ-8
C     PRINT Z BLOCK.
   62 DO 80 I=1,J,8
      K=I+7                                                              INPU1290
      DO 65 J=I,K                                                        INPU1300
      IF(Z(J))70,65,70                                                   INPU1310
   65 CONTINUE                                                           INPU1320
      GO TO 80                                                           INPU1330
   70 K=I+7                                                              INPU1340
      WRITE (6,8111)I,(Z(L),L=I,K)                                       INPU1350
   80 CONTINUE                                                           INPU1360
      GO TO 10000                                                        INPU1370
C                                                                        INPU1380
C                                                                        INPU1390
C                                                                        INPU1400
C     READ BINARY TAPE.
 1000 MZ=150                                                             INPU1410
      IWS=0                                                              INPU1420
 1003 REWIND 7
 1004 READ(7)PR(1),PR(2),N3
      NR=N3+5                                                            INPU1450
 1006 IF(PR(1)-555.0)1010,1016,1010                                      INPU1460
 1010 IWS=IWS+1                                                          INPU1470
 1011 IF(MOD(IWS,3))9902,9902,1003                                       INPU1480
 1016 IF(PR(2))1010,1018,1018                                            INPU1490
C     CHECK HERE FOR THE CORRECT CYCLE NUMBER.
 1018 IF(PK(2)-PR(2))1023,1023,1020                                      INPU1500
C     SKIP OVER, LOOK AT NEXT CYCLE.
 1020 DO 1022 L=2,NR                                                     INPU1510
 1022 READ(7)
      GO TO 1004                                                         INPU1530
 1023 READ(7)(Z(I),I=1,MZ)
C     CHECK FOR THE CORRECT PROBLEM NO.
      IF(ABS(PROB-PK(1))-.01)1024,1024,9901                              INPU1550
 1024 READ(7)(U(I),V(I),AMX(I),AIX(I),P(I),I=1,KMAXA)
      READ(7)X(0),(X(I),TAU(I),I=1,IMAX)
      READ(7)(Y(I),I=0,JMAX)
C     NOTE, INITIALIZE N1 TO 2,AND N2 TO 3
C     NOTE,N1 AND N2 ARE ONLY USED FOR BOOK-KEEPING ********
      N1=2
      N2=3
C     NOTE, IF PROBLEM NO. IS NEGATIVE, THIS IS
C     A PIC TRANSPORT, CHECK YOUR PH2 ROUTINE,
C     AND READ THE PARTICLES FROM TAPE ONTO
C     SCRATCH TAPE N1.
      IF(PROB)1,1,1034                                                   INPU1590
    1 REWIND 2
      REWIND 3
      DO 1025 I=1,N3                                                     INPU1620
```

```
      READ(7)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,N4)
      WRITE(2)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,N4)
 1025 CONTINUE
 1034 READ(7)PR(1),PR(2),PR(3)
      REWIND 2
 1036 IF(PR(1)-555.0)9904,1040,1038
 1038 IF(PR(2)-666.0)9905,1040,9905
 1040 GO TO 10
C**** END OF READ TAPE ****************************************************
C
C
C     CALCULATE MAX. GAMMA AND GAMMA/(GAMMA-1.).
C
 2000 IF(WSGX)9906,2010,2005
 2005 GAMX=1.0/(WSGX-1.0)
 2010 WSGX=(GAMX+1.0)/GAMX
      GMAXR=GAMX*WSGX
 2012 IF(WSGD)9907,2020,2015
 2015 GAMD=1.0/(WSGD-1.0)
 2020 WSGD=(GAMD+1.0)/GAMD
      GMADR=GAMD*WSGD
      GMAX=WSGD
      IF(WSGD-WSGX)2025,2030,2030
 2025 GMAX=WSGX
 2030 GO TO 40
C**** END OF R E S ********************************************************
C
C
C                ERROR
 9901 NK=1023
      GO TO 9999
 9902 NK=1011
      GO TO 9999
 9904 NK=1036
      GO TO 9999
 9905 NK=1038
      GO TO 9999
 9906 NK=2000
      GO TO 9999
 9907 NK=2012
 9999 NR=1
      CALL DUMP
C
10000 RETURN
C
C                FORMATS
 8000 FORMAT(7E10.3,I2)
 80040FORMAT(I1,71H
     1                    )
 8111 FORMAT(I4,80I4)
  END
```

```
$IBFTC CARDS   LIST,DECK,REF
      SUBROUTINE CARDS                                                  CARD0010
      DIMENSION TABLE(1),CARD(7),LABLE(1)                               CARD0020
      COMMON            TABLE                                           CARD0030
C     A 2 IN COLUMN 1, ROUTINE WILL FIX THE
C     FLOATING PT. NO.
C     A 1 IN COLUMN 1, MEANS THIS IS LAST CARD TO
C     READ IN.
      EQUIVALENCE(TABLE(1),LABLE(1))                                    CARD0050
      WRITE (6,10)                                                      CARD0070
    1 READ  (5,11)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)                  CARD0080
      WRITE (6,12)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)                  CARD0090
      DO 4 I=1,NUMWPC                                                   CARD0100
      J=LOC+I-1                                                         CARD0110
      IF(IEND-2)2,5,2                                                   CARD0120
    5 LABLE(J)=IFIX(CARD(I))                                            CARD0130
      GO TO 4                                                           CARD0140
    2 TABLE(J)=CARD(I)                                                  CARD0150
    4 CONTINUE                                                          CARD0160
      IF(IEND-1)1,3,1                                                   CARD0170
    3 RETURN                                                            CARD0180
C                FORMATS                                                CARD0190
   10 FORMAT(20H1  OIL   INPUT CARDS///)
   11 FORMAT(I1,I5,I1,0P7E9.4)                                          CARD0210
   12 FORMAT(1H I4,I7,I3,1P7E14.6)                                      CARD0220
      END                                                               CARD0230
```

```
$IBFTC SETUP   LIST,DECK,REF
      SUBROUTINE SETUP                                                  SETU0010
C     WILL ONLY GENERATE (1) MATERIAL.
C     PACKAGES MUST BE RECTANGLES.
C     ASSUMPTION OF = DX AND = DY                                       SETU0980
C     LOAD PK(4)=1.
      M=PK(4)                                                           SETU0990
C     LOAD PK(5)=RIGHT BOUNDARY OF PELLET(I).
      MA=PK(5)                                                          SETU1000
C     LOAD PK(6)=BOTTOM(J)+1 OF PELLET.
      MB=PK(6)                                                          SETU1010
C     LOAD PK(7)=TOP(J) OF PELLET.
      MC=PK(7)                                                          SETU1020
C     LOAD PK(8)=1.
      MD=PK(8)                                                          SETU1030
C     LOAD PK(9)=RIGHT(I)BOUNDARY OF TARGET.
      ME=PK(9)                                                          SETU1040
C     LOAD PK(10)=BOTTOM(J)+1 OF TARGET.
      MZ=PK(10)                                                         SETU1050
C     LOAD PK(11)=TOP(J)OF TARGET.
      N=PK(11)                                                          SETU1060
C     LOAD INITIAL DENSITY INTO Z(111).
      RHO=Z(111)                                                        SETU1070
C     LOAD INITIAL PELLET VELOCITY INTO Z(112).
      VTEF=Z(112)                                                       SETU1080
      KMAX=IMAX*JMAX+1                                                  SETU1090
      KMAXA=KMAX+1                                                      SETU1100
      JMAXA=JMAX+1                                                      SETU1110
      IMAXA=IMAX+1                                                      SETU1120
C     CLEAR ALL CELL ARRAYS.
      DO 1 K=1,KMAX                                                     SETU1130
      U(K)=0.0                                                          SETU1140
      V(K)=0.0                                                          SETU1150
      P(K)=0.0                                                          SETU1160
      AMX(K)=0.0                                                        SETU1170
      AIX(K)=0.0                                                        SETU1180
    1 CONTINUE                                                          SETU1190
      DX(1)=DX(1)                                                       SETU1200
      X(1)=DX(1)                                                        SETU1210
      WS=X(1)**2                                                        SETU1220
      PIDY=3.1415927                                                    SETU1230
      TAU(1)=WS*PIDY                                                    SETU1240
C     CALCULATE DX,X,TAU
      DO 10 I=2,IMAX                                                    SETU1250
      X(I)=X(I-1)+DX(1)                                                 SETU1260
      DX(I)=DX(1)                                                       SETU1270
      WSA=X(I)**2                                                       SETU1280
      TAU(I)=PIDY*(WSA-WS)                                              SETU1290
      WS=WSA                                                            SETU1300
   10 CONTINUE                                                          SETU1310
```

```
      Y(1)=DY(1)                                                        SETU1320
C     CALCULATE DY AND Y.
      DO 20 J=2,JMAX                                                    SETU1330
      Y(J)=Y(J-1)+DY(1)                                                 SETU1340
      DY(J)=DY(1)                                                       SETU1350
   20 CONTINUE                                                          SETU1360
      ETH=0.0                                                           SETU1370
      DO 30 I=M,MA                                                      SETU1380
      K=(MB-1)*IMAX+I+1                                                 SETU1390
C     CALCULATE MASS, AND VELOCITY OF PELLET.
      DO 40 J=MB,MC                                                     SETU1400
      AMX(K)=RHO*DY(J)*TAU(I)                                           SETU1410
      V(K)=VTEF                                                         SETU1420
C     CALCULATE TOTAL ENERGY (ETH.)
      ETH=ETH+AMX(K)*(V(K)**2)/2.0                                      SETU1430
   40 K=K+IMAX                                                          SETU1440
   30 CONTINUE                                                          SETU1450
C     CALCULATE MASS OF TARGET.
      DO 50 I=MD,ME                                                     SETU1460
      K=(MZ-1)*IMAX+I+1                                                 SETU1470
      DO 60 J=MZ,N                                                      SETU1480
      AMX(K)=RHO*DY(J)*TAU(I)                                           SETU1490
   60 K=K+IMAX                                                          SETU1500
   50 CONTINUE                                                          SETU1510
      IMAX=IMAX                                                         SETU1520
      JMAX=JMAX                                                         SETU1530
      SHELL=2.0                                                         SETU1540
      CYCLE=0.0                                                         SETU1550
      DT=0.0                                                            SETU1560
      NMAX=0                                                            SETU1570
      N1=2                                                              SETU1580
      N2=3                                                              SETU1590
      N3=0                                                              SETU1600
      N4=127                                                            SETU1610
      XMAX=X(IMAX)                                                      SETU1620
      TXMAX=XMAX*2.0                                                    SETU1630
      YMAX=Y(JMAX)                                                      SETU1640
      TYMAX=YMAX*2.0                                                    SETU1650
      REWIND  7
      WS=555.0                                                          SETU1670
C     WRITE OUTPUT FOR OIL ON TAPE.
      WRITE ( 7)WS,CYCLE,N3
      WRITE ( 7)(Z(I),I=1,150)
      WRITE ( 7)(U(I),V(I),AMX(I),AIX(I),P(I),I=1,KMAXA)
      WRITE ( 7)X(0),(X(I),TAU(I),I=1,IMAX)
      WRITE ( 7)(Y(I),I=0,JMAX)
      WS=666.0                                                          SETU1730
      WRITE ( 7)WS,WS,WS
      REWIND  7
      RETURN                                                            SETU1760
      END
```

```
$IBFTC CDT     LIST,DECK,REF
      SUBROUTINE CDT
C
C
C     ==============================================
C
C
C     CHECK COURANT CONDITION AND PARTICLE
C     VELOCITY.
C     RECORD I AND J OF ZONE WHERE DT IS BEING
C     CONTROLLED.
 3000 VEL=0.0
 3005 DO 3050 I=1,I1
 3010 K=I+1
 3015 DO 3050 J=1,I2
      I=I
      J=J
 3020 IF(AMX(K))9901,3050,3025
C
C     CALCULATE PRESSURES FROM EQUATION OF STATE(ES).
 3025 CALL ES
C
 3030 IF(ABS(P(K))-1.0E-20)3035,3035,3040
 3035 P(K)=0.0
 3040 IF(WSGX-VEL)3050,3050,3045
 3045 VEL=WSGX
 3050 K=K+IMAX
 3055 KDT=1
      UVMAX=-1.0
 3070 DO 3255 I=1,I1
 3075 K=I+1
 3095 DO 3255 J=1,I2
 3100 KP=K+IMAX
      IF(AMX(K))9901,3255,4
C     IF RHO(K) IS LESS THAN Z(138), CELL K
C     WILL BE BYPASSED FOR STABILITY CHECK.
    4 IF(AMX(K)/(TAU(I)*DY(J))-Z(138))3255,3255,3115
 3115 SIG=DX(I)
 3120 IF (DY(J)-SIG)3125,3130,3130
 3125 SIG=DY(J)
C     C=SPEED OF SOUND FOR POLYTROPIC GAS AS
C     THE SQ. ROOT OF (GAMMA*P/RHO).
C     HERE CALCULATE THE SPEED OF SOUND FOR
C     THE EQUATION OF STATE
C     AS THE SQ. ROOT OF DP/DRHO.
 3130 IF(Z(148))4000,4000,4001
 4000 WS=SQRT(GMAX*TAU(I)*DY(J)*ABS(P(K))/(AMX(K)))
      GO TO 3205
 4001 WSA=ABS(P(K))*1.E+4
      WS=Z(148)+Z(149)*(WSA**Z(150))
```

```
      WS=WS*1.E-3                                                       CDT 1340
 3205 WS=WS/SIG                                                         CDT 1350
 3210 IF(UVMAX-WS)3215,3220,3220                                        CDT 1360
 3215 N10=I                                                             CDT 1370
      N11=J                                                             CDT 1380
      UVMAX=WS                                                          CDT 1390
 3220 IF(NMAX)1,1,2                                                     CDT 1400
C     EULERIAN CHECK FOR RADIAL PARTICLE VELOCITY.
    1 CONTINUE                                                          CDT 1410
    3 WS=ABS(U(K))/TAU(I)*X(I)/.5*PIDY                                  CDT 1420
      GO TO 3225                                                        CDT 1430
C     PIC CHECK FOR RADIAL PARTICLE VELOCITY.
    2 WS=ABS(U(K))/DX(I)                                                CDT 1440
 3225 IF(UVMAX-WS)3230,3235,3235                                        CDT 1450
 3230 UVMAX=WS                                                          CDT 1460
      N10=I                                                             CDT 1470
      N11=J                                                             CDT 1480
 3235 WS=ABS(V(K))/DY(J)                                                CDT 1490
 3240 IF(UVMAX-WS)3245,3250,3250                                        CDT 1500
 3245 N10=I                                                             CDT 1510
      N11=J                                                             CDT 1520
      UVMAX=WS                                                          CDT 1530
 3250 CONTINUE                                                          CDT 1540
 3255 K=K+IMAX                                                          CDT 1550
      IF(UVMAX)9912,9912,3260
C     FOR OPTIONS ON CABLN, CHECK
C     SECTION 3.4 IN GAMD-5580.
 3260 IF(CABLN)90,91,3300                                               CDT 1560
   90 DT=.5/VEL/UVMAX*Z(139)                                            CDT 1570
      GO TO 3295                                                        CDT 1580
   91 WS=UVMAX*DT                                                       CDT 1590
      WSA=0.5/VEL                                                       CDT 1600
 3265 IF(FFA-WSA)3276,3276,3270                                         CDT 1610
 3270 FFA=WSA                                                           CDT 1620
 3276 IF(WS-FFA)3285,3300,3280                                          CDT 1630
 3280 DT=DT/WS*FFB/0.9                                                  CDT 1640
      GO TO 3295                                                        CDT 1650
 3285 IF(WS-FFB)3290,3290,3300                                          CDT 1660
 3290 DT=DT*FFA/WS*0.9                                                  CDT 1670
 3295 KDT=0                                                             CDT 1680
C     INTEGRATE THE TIME AND CYCLE COUNTER.
 3300 T=T+DTNA                                                          CDT 1690
   85 IF(DTRAD)9911,80,81                                               CDT 1700
   80 NR=NRN                                                            CDT 1710
   84 WS=NR                                                             CDT 1720
      TRAD=DT/WS                                                        CDT 1730
      GO TO 82                                                          CDT 1740
   81 IWS=DT/DTRAD                                                      CDT 1750
      NR=IWS+1                                                          CDT 1760
   83 IF(NR-NRM)84,84,80                                                CDT 1770
```

```
   82 NC=NC+1
      CYCLE=NC                                                    CDT 1780
      NPC=NPC+1                                                   CDT 1790
 3305 IF(T)9909,3320,3310                                         CDT 1800
 3310 IF(KDT)9910,3315,3320                                       CDT 1810
 3315 WRITE (6,8000)T,DTNA,DT                                     CDT 1820
 3320 DTNA=DT                                                     CDT 1830
      GO TO 3325                                                  CDT 1840
C                NEGATIVE MASS                                    CDT 1850
 9901 NK=3020                                                     CDT 1860
      GO TO 9999                                                  CDT 1870
 9909 NK=3305                                                     CDT 1880
      GO TO 9999                                                  CDT 1890
 9910 NK=3310                                                     CDT 1900
      GO TO 9999                                                  CDT 1910
C     THE DT WILL BE 0. OR NEGATIVE ,STOP                         CDT 1920
 9912 NK=1
      GO TO 9999
 9911 NK=85
 9999 NR=2                                                        CDT 1930
      CALL DUMP                                                   CDT 1940
 3325 RETURN                                                      CDT 1950
80000FORMAT (17HOCHANGE DT ... T=1PE9.3,11H      DT(N)=1PE9.3,13H      DTCDT 1960
    1(N+1)=1PE9.3)                                                CDT 1970
      END                                                         CDT 1980
                                                                  CDT 1990
```

```
$IBFTC PH1     LIST,DECK,REF
      SUBROUTINE PH1                                                    PH1 0010
C                                                                       PH1 0900
C     VELOCITIES, ENERGIES, PRESSURES ARE AT THE
C     CENTER OF THE CELL.
C     (2) PASSES THRU PH1 ARE REQUIRED. NO
C     MASS IS MOVED IN PH1.
C     ****** NOTE 1 MATERIAL ONLY (X)) ************                    PH1 0980
C                                                                       PH1 0990
C                                                                       PH1 1000
C                                                                       PH1 1010
C     ============================================                      PH1 1020
C                                                                       PH1 1030
C                                                                       PH1 1040
      NRT=0                                                             PH1 1050
      NRC=0                                                             PH1 1060
      UU=1.E+15                                                         PH1 1070
      UT=0.0                                                            PH1 1080
C     YOU WILL GET BACK HERE IF AIX WAS LESS
C     THAN 0. AND PROVIDED SN=0.
 8000 VEL=1.0                                                           PH1 1090
C     INITIALIZE MID-POINTS OF FIRST AND SECOND
C     CELL IN R DIRECTION.
 3301 RC=DX(1)/2.0                                                      PH1 1100
      RR=(X(1)+X(2))/2.0                                                PH1 1110
 3304 K=2                                                               PH1 1120
C     AXIS OF SYMMETRY BOUNDARY CONDITIONS.
      DO 3302 J=1,JMAX                                                  PH1 1130
      PL(J)=P(K)                                                        PH1 1140
      UL(J)=0.0                                                         PH1 1150
 3302 K=K+IMAX                                                          PH1 1160
C     FIRST PASS THRU, CALCULATE U AND V AT
C     CYCLE N+1, AND THE WORK TERMS USING U AND V
C     FROM CYCLE N.
C     SECOND PASS THRU, CALCULATE ONLY THE
C     CONTRIBUTION TO THE CHANGE IN INTERNAL ENERGY
C     FROM WORK TERMS EVALUATED FROM U AND V
C     AT CYCLE N+1.
      DO 3360 I=1,I1                                                    PH1 1170
      K=I+1                                                             PH1 1180
      IF(CVIS)7002,7003,7003                                            PH1 1190
C     BOTTOM BOUNDARY IS TRANSMITTIVE.
 7002 VBLO=V(K)                                                         PH1 1200
      PBLO=0.0                                                          PH1 1210
      GO TO 7004                                                        PH1 1220
C     BOTTOM BOUNDARY IS REFLECTIVE.
 7003 VBLO=0.0                                                          PH1 1230
      PBLO=P(K)                                                         PH1 1240
 7004 TAUDTS=TAU(I)*DT                                                  PH1 1250
C     I1= MAX.(I) OF DISTURBANCE IN R DIRECTION.
```

```
C     I2= MAX(J) OF DISTURBANCE IN Z DIRECTION.
C     DO LOOP IN J DIRECTION
      DO 3348 J=1,I2                                                    PH1 1260
      PIDTS=1.0/(PIDY*DT*DY(J))                                         PH1 1270
C     K= INDEX OF CELL IN QUESTION.
C     N= INDEX OF CELL ABOVE.
      N=K+IMAX                                                          PH1 1280
 3305 IF(AMX(K))9902,3340,3306                                          PH1 1290
 3306 IF(IMAX-I)9903,3311,3310                                          PH1 1300
 3310 IF(AMX(K+1))9904,3312,3314                                        PH1 1310
C     WE ARE AT THE RIGHT BOUNDARY, SET PRESSURE
C     GRADIENT TO 0. IN R DIRECTION, MODIFY ETH.
C     FOR RIGHT BOUNDARY BEING TRANSMITTIVE.
 3311 PRR=PL(J)                                                         PH1 1320
 3307 ETH=ETH-PRR*U(K)/PIDTS*RC                                         PH1 1330
      GO TO 3313                                                        PH1 1340
C     RIGHT BOUNDARY CONDITION FOR THE MOMENTUM EQ.
C     ADJACENT TO EMPTY CELL.
 3312 PRR=0.0                                                           PH1 1350
 3313 URR=RC*U(K)                                                       PH1 1360
      GO TO 3316                                                        PH1 1370
C     CALCULATE PRESSURE AT INTERFACE(I) AND
C     (RU) FOR WORK TERM.
 3314 PRR=(P(K)+P(K+1))/2.0                                             PH1 1380
 3315 URR=(U(K)*RC+U(K+1)*RR)/2.0                                       PH1 1390
 3316 IF(JMAX-J)9905,3318,3320                                          PH1 1400
C     SET PRESSURE GRADIENT TO 0. THIS IS FOR TOP
C     BOUNDARY BEING TRANSMITTIVE.
 3318 PABOVE=PBLO                                                       PH1 1410
C     MODIFY ETH FOR TOP BOUNDARY CONDITION.
 3319 ETH=ETH-PABOVE*V(K)/2.0*TAUDTS                                    PH1 1420
      GO TO 3323                                                        PH1 1430
 3320 IF(AMX(N))9906,3322,3324                                          PH1 1440
C     TOP BOUNDARY CONDITION (EMPTY CELL ABOVE.)
C     TOP BOUNDARY CONDITION FOR VELOCITY (EMPTY CELL ABOVE).
 3322 PABOVE=0.0                                                        PH1 1450
 3323 VABOVE=V(K)                                                       PH1 1460
      GO TO 3328                                                        PH1 1470
C     CALCULATE PRESSURE AT INTERFACE(J)
 3324 PABOVE=(P(K)+P(N))/2.0                                            PH1 1480
      IF(CVIS)7001,3325,3325                                            PH1 1490
 7001 IF(1-J)3325,7000,9905                                             PH1 1500
C     BOTTOM BOUNDARY IS TRANSMITTIVE, SET PRESSURE
C     GRADIENT TO 0.
C     AND MODIFY ETH.
 7000 PBLO=PABOVE                                                       PH1 1510
      ETH=ETH+PBLO*V(K)/2.0*TAUDTS                                      PH1 1520
C     VELOCITY AT INTERFACE(J)
 3325 VABOVE=(V(K)+V(N))/2.0                                            PH1 1530
 3328 IF(VEL)9907,3404,3400                                             PH1 1540
```

```
C     COMPUTE DELTA U AND DELTA V.
 3400 V(K)=V(K)+(PBLO-PABOVE)*TAUDTS/(AMX(K))                          PHI 1550
      IF(ABS(V(K))-1.E-08)3401,3401,3402                               PHI 1560
 3401 V(K)=0.0                                                         PHI 1570
 3402 U(K)=U(K)+(PL(J)-PRR)/(AMX(K))*RC/PIDTS*2.0                      PHI 1580
      IF(ABS(U(K))-1.E-08)3403,3403,3404                               PHI 1590
 3403 U(K)=0.0                                                         PHI 1600
C     CHECK FOR ADVANCING COUNTERS OF THE ACTIVE
C     GRID IN THE R DIRECTION.
 3404 IF(I-I1)6016,6005,6005                                           PHI 1610
 6005 IF(U(K))6605,6606,6605                                           PHI 1620
 6605 NRC=1                                                            PHI 1630
 6606 IF(V(K))6607,6004,6607                                           PHI 1640
 6607 NRC=1                                                            PHI 1650
 6004 IF(AIX(K))6015,6016,6015                                         PHI 1660
 6015 NRC=1                                                            PHI 1670
 6016 CONTINUE                                                         PHI 1680
C     HERE CALCULATE CHANGE IN INTERNAL ENERGY
C     DUE TO WORK TERMS ONLY.
      WS=(VBLO-VABOVE)*TAUDTS/2.0*P(K)                                 PHI 1690
      RHO=WS+(UL(J)-URR)/PIDTS*P(K)                                    PHI 1700
C     CONVERT TO SPECIFIC INTERNAL ENERGY.
 3332 WSX=AIX(K)+RHO/AMX(K)                                            PHI 1710
      GO TO 1000                                                       PHI 1720
C     CHECK FOR NEGATIVE INTERNAL ENERGIES.
 1000 IF(WSX)1011,1001,1001                                            PHI 1730
 1001 AIX(K)=WSX                                                       PHI 1740
      GO TO 3342                                                       PHI 1750
 1011 UT=1.0                                                           PHI 1760
C     COMPUTE NEW DT(STORE IN UU) ASSUMING
C     THAT DI/DT WILL BE THE SAME FOR A SMALLER
C     TIME STEP, THE NEW DT IS CHOSEN SUCH
C     THAT AIX(AT N+1)=2/3 OF AIX(N).
      WSA=2.0*AIX(K)/3.0*DT/(AIX(K)-WSX)                               PHI 1770
 1013 IF(WSA-UU)1014,1001,1001                                         PHI 1780
 1014 UU=WSA                                                           PHI 1790
      GO TO 1001                                                       PHI 1800
C     CELL (K) IS EMPTY, SET INTERFACE QUANTITIES,
C     ASSUMING CELL TO THE RIGHT AND TOP ARE
C     NOT VOID.
 3340 PRR=0.0                                                          PHI 1810
      URR=U(K+1)*RR                                                    PHI 1820
      PABOVE=0.0                                                       PHI 1830
      VABOVE=V(N)                                                      PHI 1840
C     SET RIGHT QUANTITIES TO THE LEFT (FOR NEXT
C     COLUMN SWEEP) AND SET ABOVE QUANTITIES TO
C     BELOW FOR NEXT CELL ABOVE.
 3342 VBLO=VABOVE                                                      PHI 1850
      PL(J)=PRR                                                        PHI 1860
      UL(J)=URR                                                        PHI 1870
```

```
      K=N
 3348 PBLO=PABOVE
      LL=K-IMAX
C     CHECK FOR ADVANCING COUNTERS OF THE ACTIVE
C     GRID IN Z DIRECTION.
      IF(U(LL))6000,6001,6000
 6000 NRT=1
 6001 IF(V(LL))6002,6003,6002
 6002 NRT=1
 6003 IF(AIX(LL))6017,6018,6017
 6017 NRT=1
 6018 CONTINUE
 3355 RC=RR
      RR=(X(I+1)+X(I+2))/2.0
 3360 CONTINUE
 3361 IF(VEL)9911,10000,3363
 3363 VEL=0.0
      GO TO 3301
C                ERROR
 9902 NK=3305
      GO TO 9999
 9903 NK=3306
      GO TO 9999
 9904 NK=3310
      GO TO 9999
 9905 NK=3316
      GO TO 9999
 9906 NK=3320
      GO TO 9999
 9907 NK=3328
      GO TO 9999
 9911 NK=3361
 9999 NR=3
      CALL DUMP
C     IF SN(NOT=0.) ANY NEGATIVE ENERGIES WILL
C     REMAIN. IF=0, CODE WILL TRY ANOTHER PASS
C     WITH A SMALLER DT.
10000 IF(SN)7030,7031,7030
 7031 IF(UT)7020,7030,7010
C     NEGATIVE ENERGIES HAVE OCCURED, INTEGRATE
C     BACK TO CYCLE N WITH (-DT).
 7010 UT=-1.0
      DT=-DT
C     YOU NOW HAVE INTEGRATED BACK TO CYC   N. NOW
C     INTEGRATE TO CYCLE N+1 WITH NEW DT(STORED IN UU).
      GO TO 8000
 7020 UT=0.0
      DT=UU
      NR=DT/TRAD+1.0
      WS=NR
```

```
      TRAD=DT/WS
      DTNA=DT
      GO TO 8000
C     INCREASE ACTIVE GRID COUNTERS IF NEEDED.
 7030 I1=I1+NRC
      I2=I2+NRT
      IF(I1-IMAX)6100,6100,6200
 6200 I1=IMAX
 6100 IF(I2-JMAX)6201,6201,6202
 6202 I2=JMAX
 6201 RETURN
      END
```

```
$IBFTC PH2     LIST,DECK,REF
      SUBROUTINE PH2                                                    PH2 0010
C     Z(110)= CRITICAL ENERGY( BETWEEN GAS AND CONDENSED STATE)
C     Z(111)= INITIAL DENSITY
C     Z(112)= INITIAL VELOCITY OF PELLET
C     Z(113)= EPSILONICS FOR EMPTYING PELLET ON BASIS OF VELOCITY
C     TOZONE = MINIMUM DENSITY FOR MASS FLOW
C                                                                       PH2 0900
C                                                                       PH2 0980
C     AMPY=MASS ACROSS TOP BOUNDARY OF CELL
C     AMUT=RADIAL MOMENTA OF THIS MASS
C     AMVT=AXIAL MOMENTA OF THIS MASS
C     DELET=TOTAL SPECIFIC ENERGY OF THIS MASS
C     AMMP=MASS ACROSS RIGHT BOUNDARY OF CELL
C     AMUR=RADIAL MOMENTA OF THIS MASS
C     AMVR=AXIAL MOMENTA OF THIS MASS
C     DELER=TOTAL SPECIFIC ENERGY OF THIS MASS
C     AMMY=MASS ACROSS BOTTOM BOUNDARY OF CELL
C     AMMU=RADIAL MOMENTA OF THIS MASS
C     AMMV=AXIAL MOMENTA OF THIS MASS
C     DELEB=TOTAL SPECIFIC ENERGY OF THIS MASS
C     GAMC=MASS ACROSS LEFT BOUNDARY OF CELL
C     FLEFT=RADIAL MOMENTA OF THIS MASS
C     YAMC=AXIAL MOMENTA OF THIS MASS
C     SIGC=TOTAL SPECIFIC ENERGY OF THIS MASS
C     ==========================================                        PH2 0990
C                                                                       PH2 1010
      NRT=0                                                             PH2 1020
      NRC=0                                                             PH2 1030
      REZ=0.0                                                           PH2 1040
      CALL SLITE (0)                                                    PH2 1050
      PIDTS=1.0/(PIDY*DT)                                               PH2 1060
  101 DO 103 J=1,JMAX                                                   PH2 1070
  102 GAMC(J)=0.0                                                       PH2 1080
      FLEFT(J)=0.0                                                      PH2 1090
      YAMC(J)=0.0                                                       PH2 1100
      SIGC(J)=0.0                                                       PH2 1110
  103 CONTINUE                                                          PH2 1120
  104 DO 547 I=1,I1                                                     PH2 1130
      J=1                                                               PH2 1140
  105 K=I+1                                                             PH2 1150
   80 IF(AMX(K))9900,97,81                                              PH2 1160
   81 IF(V(K))82,97,97                                                  PH2 1170
   97 AMMV=0.0                                                          PH2 1180
      GO TO 98                                                          PH2 1190
   82 AMMY=AMX(K)*V(K)*DT/DY(J)                                         PH2 1200
   83 IF(AMMY+AMX(K))84,85,85                                           PH2 1210
   84 AMMY=-AMX(K)                                                      PH2 1220
   85 IF(CVIS)106,99,99                                                 PH2 1230
C     BOTTOM BOUNDARY IS TRANSMITTIVE, MATERIAL IS MOVING
```

```
C       OUT, REMOVE ITS ENERGY FROM ETH.
  106 AMMU=AMMY*U(K)                                                    PH2 1240
      AMMV=AMMY*V(K)                                                    PH2 1250
      DELEB=AIX(K)+(U(K)**2+V(K)**2)/2.0                                PH2 1260
      WS=(U(K)**2+V(K)**2)/2.0                                          PH2 1270
      ETH=ETH+AMMY*(AIX(K)+WS)                                          PH2 1280
      GO TO 107                                                         PH2 1290
C       BOTTOM BOUNDARY IS REFLECTIVE, NET MOMENTA CHANGE
C       IN Z DIRECTION IS 2 MV.
   99 AMMV=2.0*AMMY*V(K)                                                PH2 1300
   98 AMMY=0.0                                                          PH2 1310
      AMMU=0.0                                                          PH2 1320
      DELEB=0.0                                                         PH2 1330
C       BEGIN DO LOOP IN J(Z) DIRECTION.
  107 DO 546 J=1,I2                                                     PH2 1340
  108 L=K+IMAX                                                          PH2 1350
      I=I                                                               PH2 1360
      J=J                                                               PH2 1370
      AREA=0.0                                                          PH2 1380
      VEL=0.0                                                           PH2 1390
      FS=0.0                                                            PH2 1400
  210 IF(JMAX-J)211,211,207                                             PH2 1410
  211 VEL=1.0                                                           PH2 1420
      GO TO 208                                                         PH2 1430
  207 IF(AMX(L))215,215,214                                             PH2 1440
  214 IF(AMX(K))216,216,209                                             PH2 1450
  216 VABOVE=V(L)                                                       PH2 1460
      GO TO 212                                                         PH2 1470
  215 IF(AMX(K))205,205,208                                             PH2 1480
  205 VABOVE=0.0                                                        PH2 1490
      GO TO 212                                                         PH2 1500
  208 VABOVE=V(K)                                                       PH2 1510
      GO TO 212                                                         PH2 1520
  209 VABOVE=(V(K)+V(L))/2.0                                            PH2 1530
  212 CONTINUE                                                          PH2 1540
      I=I                                                               PH2 1550
      J=J                                                               PH2 1560
      FS=0.0                                                            PH2 1570
  404 IF(IMAX-I)412,412,405                                             PH2 1580
  405 IF(AMX(K+1))411,411,409                                           PH2 1590
  409 IF(AMX(K))410,410,407                                             PH2 1600
  410 URR=U(K+1)                                                        PH2 1610
      GO TO 408                                                         PH2 1620
  411 IF(AMX(K))403,403,406                                             PH2 1630
  403 URR=0.0                                                           PH2 1640
      GO TO 408                                                         PH2 1650
C       WE ARE AT THE RIGHT BOUNDARY OF THE GRID, THE
C       BOUNDARY CONDITION HERE IS TRANSMITTIVE.
  412 FS=1.0                                                            PH2 1660
  406 URR=U(K)                                                          PH2 1670
```

```
      GO TO 408
  407 URR=(U(K)+U(K+1))/2.0
  408 CONTINUE
  109 IF(AREA)9901,301,547
  301 IF(VABOVE)300,304,302
  302 IF(AMX(K))9900,304,8800
 8800 IF(J-1)9900,303,8801
 8801 KP=K-IMAX
      IF(AMX(KP))9900,8803,303
C     A CHECK HERE TO INSURE THAT THE BOTTOM ZONES
C     OF THE PROJECTILE EMPTY (FOR HYPERVELOCITY) UP UNTIL
C     THE INITIAL VELOCITY CHANGES DUE TO THE SHOCK.
 8803 IF(ABS(VABOVE-Z(112))/Z(112)-Z(113))306,303,303
  303 M=K
      JJ=J
      GO TO 307
  304 AMPY=0.0
  308 AMUT=0.0
      AMVT=0.0
      DELET=0.0
      GO TO 501
  300 IF(VEL)9901,305,304
  305 IF(AMX(L))9903,304,306
  306 M=L
      JJ=J+1
  307 IF(VEL)6130,6130,6140
 6130 WSA=(V(K)+V(L))/2.0
      WSB=1.0+(V(L)-V(K))/(DY(JJ)*SBOUND)*DT
      VABOVE=WSA/WSB
C     CALCULATE THE MASS FLUX AT THE TOP OF CELL K.
 6140 AMPY=AMX(M)*VABOVE/DY(JJ)*DT
  501 IF(URR)500,504,502
  502 IF(AMX(K))9900,504,503
  503 M=K
      N=I
      GO TO 508
  504 AMMP=0.0
      AMUR=0.0
      AMVR=0.0
      DELER=0.0
      GO TO 1
  500 IF(FS)9905,506,504
  506 IF(AMX(K+1))9904,504,507
  507 M=K+1
      N=I+1
  508 IF(FS)6100,6100,6110
 6100 WSA=(U(K)+U(K+1))/2.0
      WSB=1.0+(U(K+1)-U(K))/(DX(N)*SBOUND)*DT
      URR=WSA/WSB
C     CALCULATE THE MASS FLUX AT THE RIGHT OF CELL K.
```

```
 6110 DEN=AMX(M)/TAU(N)                                                 PH2 2130
      AMMP=DEN/PIDTS*X(I)/.5*URR
    1 IF(AMMP)16,74,8820                                                PH2 2150
 8820 IF(GAMC(J))74,74,8821                                             PH2 2160
 8821 IF(FS)6120,6120,74                                                PH2 2170
 6120 IF(AMX(K+1))9903,8822,74                                          PH2 2180
 8822 IF(AMX(K)/(TAU(I)*DY(J))-Z(111))8823,74,74                        PH2 2190
 8823 IF(AIX(K)-Z(110))8824,74,74                                       PH2 2200
 8824 WS=GAMC(J)+AMX(K)-TAU(I)*DY(J)*Z(111)                             PH2 2210
      IF(WS)8826,8826,8825                                              PH2 2220
 8825 AMMP=WS                                                           PH2 2230
      GO TO 74                                                          PH2 2240
 8826 AMMP=0.0                                                          PH2 2250
   74 JTAG=0                                                            PH2 2260
C     BEGIN CHECKING TO SEE IF THEIR IS ANY
C     PREFERENTIAL MASS FLUX BECAUSE OF CHOICE OF
C     INDEXING DIRECTION.
    2 IF(AMPY)3,4,4                                                     PH2 2270
C     TOP FLUX IS INTO CELL K.
    3 ITAG=1                                                            PH2 2280
      WSB=AMPY                                                          PH2 2290
      AMPY=0.0                                                          PH2 2300
      GO TO 64                                                          PH2 2310
    4 ITAG=0                                                            PH2 2320
   64 IF(AMMY)9,5,5                                                     PH2 2330
C     BOTTOM FLUX IS INTO CELL K.
    5 IF(GAMC(J))7,6,6                                                  PH2 2340
C     LEFT FLUX IS INTO CELLK.
    6 WS=AMX(K)                                                         PH2 2350
      GO TO 11                                                          PH2 2360
C     LEFT FLUX IS OUT.
    7 WS=AMX(K)+GAMC(J)                                                 PH2 2370
      GO TO 11                                                          PH2 2380
C     BOTTOM FLUX IS OUT OF CELL K.
    9 IF(GAMC(J))10,8,8                                                 PH2 2390
C     LEFT FLUX IS INTO CELL K.
    8 WS=AMX(K)+AMMY                                                    PH2 2400
      GO TO 11                                                          PH2 2410
C     LEFT FLUX IS OUT OF CELL K.
   10 WS=AMX(K)+GAMC(J)+AMMY                                            PH2 2420
   11 WSA=AMPY+AMMP                                                     PH2 2430
   12 IF(WSA-WS)75,75,13                                                PH2 2440
C     CHANGE TOP AND RIGHT FLUX TO BE THE
C     OLD FLUX TIMES THE MASS OF THE CELL/THE SUM
C     OF THE OLD FLUXES.
   13 AMPY=AMPY*WS/WSA                                                  PH2 2450
      AMMP=AMMP*WS/WSA                                                  PH2 2460
   75 IF(JTAG)14,73,14                                                  PH2 2470
   73 WSC=AMMP                                                          PH2 2480
   14 IF(ITAG)15,7000,15                                                PH2 2490
```

```
   15 AMPY=WSB
      ITAG=0
C     GO CHECK CELL ABOVE.
      GO TO 40
C     RIGHT FLUX IS INTO CELL K.
   16 IF(FS)76,17,76
   76 WSC=AMMP
C     I=IMAX, SO CHECK CELL ABOVE K.
      GO TO 40
   17 IF(I+1-IMAX)19,18,9908
   18 URRR=U(K+1)/2.0
      GO TO 20
   19 URRR=(U(K+1)+U(K+2))/2.0
   20 IF(URRR)39,39,21
C     FLUX IS OUT OF THE RIGHT OF CELL(K+1).
   21 URRR=URRR/TAU(I+1)*AMX(K+1)/PIDTS*X(I+1)/.5
   22 IF(J-1)9909,23,24
   23 VBLO=V(K+1)/2.0
      GO TO 26
   24 KP=K+1-IMAX
      VBLO=(V(K+1)+V(KP))/2.0
   26 IF(VBLO)25,38,38
C     FLUX IS OUT OF THE BOTTOM OF CELL(K+1).
   25 VBLO=AMX(K+1)/DY(J)*VBLO*DT
   27 IF(VEL)28,29,28
   28 VAB=V(K+1)/2.0
      GO TO 31
   29 KP=K+IMAX+1
      VAB=(V(K+1)+V(KP))/2.0
   31 IF(VAB)36,36,30
C     FLUX IS OUT OF TOP.
   30 VAB=AMX(K+1)/DY(J)*VAB*DT
   32 WS=AMX(K+1)
   33 WSA=URRR-AMMP-VBLO+VAB
   34 IF(WSA-WS)77,77,35
   35 AMMP=AMMP*WS/WSA
   77 JTAG=1
      WSC=AMMP
      AMMP=0.0
      GO TO 2
C     FLUX AT TOP IS INTO CELL (K+1).
   36 WS=AMX(K+1
   37 WSA=URRR-AMMP-VBLO
      GO TO 34
C     FLUX IS IN FROM BOTTOM INTO CELL (K+1).
   38 VBLO=0.0
      GO TO 27
C     FLUX IS INTO CELL (K+1) FROM RIGHT.
   39 URRR=0.0
      GO TO 22
```

```
C       RIGHT FLUX OUT OF CELL (K) IS POSITIVE AND TOP
C       FLUX IS COMING INTO CELL (K) FROM (K+IMAX).
   40 IF(VEL)7000,41,7000                                          PH2 2910
   41 IF(FS)42,43,42                                               PH2 2920
C       WE ARE AT THE RIGHT BOUNDARY OF THE GRID.
   42 KP=K+IMAX                                                    PH2 2930
      URT=U(KP)/2.0                                                PH2 2940
      GO TO 45                                                     PH2 2950
   43 KP=K+IMAX                                                    PH2 2960
      URT=(U(KP)+U(KP+1))/2.0                                      PH2 2970
   45 IF(URT)46,46,70                                              PH2 2980
C       FLUX AT RIGHT (CELL M) IS NEGATIVE.
   46 URT=0.0                                                      PH2 2990
      GO TO 47                                                     PH2 3000
   70 KP=K+IMAX                                                    PH2 3010
      URT=URT/TAU(I)*AMX(KP)/PIDTS*X(I)/.5
C       FLUX AT RIGHT (CELL M) IS POSITIVE.
   47 IF(J+1-JMAX)48,49,9910                                       PH2 3030
   48 KP=K+IMAX                                                    PH2 3040
      KL=KP+IMAX                                                   PH2 3050
      VABT=(V(KP)+V(KL))/2.0                                       PH2 3060
      GO TO 51                                                     PH2 3070
   49 KP=K+IMAX                                                    PH2 3080
      KL=KP+IMAX                                                   PH2 3090
      VABT=V(KP)/2.0                                               PH2 3100
   51 IF(VABT)8810,71,72                                           PH2 3110
C       FLUX IS IN FROM TOP OF CELL M.
 8810 IF(AMX(K))9903,8811,71                                       PH2 3120
C       CHECK FOR SOLID OR VAPOR.
 8811 IF(AMX(KP)/(TAU(I)*DY(J+1))-Z(111))8812,71,71                PH2 3130
 8812 IF(AIX(KP)-Z(110))8813,71,71                                 PH2 3140
 8813 VABT=VABT*AMX(KL)/DY(J+2)*DT                                 PH2 3150
 8814 WS=-VABT+AMX(KP)-TAU(I)*DY(J+1)*Z(111)                       PH2 3160
 8815 IF(WS)8817,8817,8816                                         PH2 3170
 8816 AMPY=-WS                                                     PH2 3180
      GO TO 71                                                     PH2 3190
 8817 AMPY=0.0                                                     PH2 3200
   71 VABT=0.0                                                     PH2 3210
      GO TO 60                                                     PH2 3220
   72 VABT=VABT*AMX(KP)/DY(J+1)*DT                                 PH2 3230
   52 IF(GAMC(J+1))54,53,53                                        PH2 3240
   53 WS=AMX(KP)                                                   PH2 3250
      GO TO 55                                                     PH2 3260
   54 WS=AMX(KP)+GAMC(J+1)                                         PH2 3270
   55 WSA=VABT-AMPY+URT                                            PH2 3280
      GO TO 57                                                     PH2 3290
   60 IF(GAMC(J+1))61,61,59                                        PH2 3300
   61 WS=AMX(KP)+GAMC(J+1)                                         PH2 3310
      GO TO 58                                                     PH2 3320
   59 WS=AMX(KP)                                                   PH2 3330
```

```
   58 WSA=-AMPY+URT                                              PH2 3340
   57 IF(WSA-WS)7000,7000,56                                     PH2 3350
   56 AMPY=AMPY*WS/WSA                                           PH2 3360
      GO TO 7000                                                 PH2 3370
 7000 AMMP=WSC                                                   PH2 3380
  309 IF(AMPY)8834,8831,8833                                     PH2 3390
 8833 IF(JMAX-J)9911,318,8835                                    PH2 3400
 8835 KP=K+IMAX                                                  PH2 3410
 8836 IF(AMX(KP))9900,8837,318                                   PH2 3420
C     **** NOTE *****************************************************
C     ACROSS FREE SURFACE, HOLD UP MASS FLUX
C      UNLESS THIS MASS PRODUCES A DENSITY GREATER THAN TOZONE.
C     ***************************************************************
 8837 IF(AMPY/(TAU(I)*DY(J))-TOZONE)8838,318,318                 PH2 3430
 8838 AMPY=0.0                                                   PH2 3440
      GO TO 8831                                                 PH2 3450
 8834 IF(J-1)9911,325,8839                                       PH2 3460
 8839 IF(AMX(K))9900,8840,325                                    PH2 3470
 8840 IF(-AMPY/(TAU(I)*DY(J))-TOZONE)8841,325,325
 8841 AMPY=0.0                                                   PH2 3490
      GO TO 8831                                                 PH2 3500
  318 DELM=GAMC(J)+AMMY-AMPY                                     PH2 3510
  322 IF(VEL)9901,324,323                                        PH2 3520
  323 WS=U(K)**2+V(K)**2                                         PH2 3530
C     MATERIAL HAS LEFT THE TOP, TRIGGER REZONE
C     FLAG, REMOVE ITS ENERGY FROM ETH(TOTAL ENERGY OF SYSTEM).
      ETH=ETH-AMPY*(AIX(K)+WS/2.0)                               PH2 3540
      IF(AMPY/(TAU(I)*DY(J))-TOZONE)324,324,6900                 PH2 3550
 6900 REZ=1.0                                                    PH2 3560
  324 AMUT=AMPY*U(K)                                             PH2 3570
      AMVT=AMPY*V(K)                                             PH2 3580
      GO TO 326                                                  PH2 3590
  325 CONTINUE                                                   PH2 3600
 8831 AMUT=AMPY*U(L)                                             PH2 3610
      AMVT=AMPY*V(L)                                             PH2 3620
      DELM=GAMC(J)-AMPY+AMMY                                     PH2 3630
  326 IF(AMPY)327,328,328                                        PH2 3640
  327 DELET=AIX(L)+(U(L)**2+V(L)**2)/2.0                         PH2 3650
      GO TO 333                                                  PH2 3660
  328 IF(AMMY)329,330,330                                        PH2 3670
  329 DELET=DELEB                                                PH2 3680
      GO TO 333                                                  PH2 3690
  330 IF(GAMC(J))331,332,332                                     PH2 3700
  331 DELET=SIGC(J)                                              PH2 3710
      GO TO 333                                                  PH2 3720
  332 DELET=AIX(K)+(U(K)**2+V(K)**2)/2.0                         PH2 3730
C     SUM UP RADIAL MOMENTA FOR ALL FLUXES EXCEPT
C     THE RIGHT AND STORE IN SIGMU.
  333 SIGMU=FLEFT(J)+AMMU-AMUT                                   PH2 3740
C     SUM UP AXIAL MOMENTA FOR ALL FLUXES EXCEPT THE
```

```
C       RIGHT AND STORE IN SIGMV.
        SIGMV=YAMC(J)+AMMV-AMVT                                          PH2 3750
C       SUM UP TOTAL ENERGY CARRIED BY THESE FLUXES
C       EXCEPT THE RIGHT FLUX AND STORE IN DELEK.
        DELEK=GAMC(J)*SIGC(J)+AMMY*DELEB-AMPY*DELET                      PH2 3760
  509   IF(AMMP)8843,518,8844                                            PH2 3770
 8844   IF(IMAX-I)9911,518,8845                                          PH2 3780
 8845   IF(AMX(K+1))9900,8846,518                                        PH2 3790
 8846   IF(AMMP/(TAU(I)*DY(J))-TOZONE)8847,518,518                       PH2 3800
 8847   AMMP=0.0                                                         PH2 3810
        GO TO 518                                                        PH2 3820
 8843   IF(I-1)9911,512,8848                                             PH2 3830
 8848   IF(AMX(K))9900,8849,512                                          PH2 3840
 8849   IF(-AMMP/(TAU(I)*DY(J))-TOZONE)8850,512,512
 8850   AMMP=0.0                                                         PH2 3860
        GO TO 518                                                        PH2 3870
  512   DELM=DELM-AMMP+AMX(K)                                            PH2 3880
  513   CONTINUE                                                         PH2 3890
  514   CONTINUE                                                         PH2 3900
 8828   AMUR=AMMP*U(K+1)                                                 PH2 3910
        AMVR=AMMP*V(K+1)                                                 PH2 3920
        GO TO 525                                                        PH2 3930
  518   DELM=DELM-AMMP+AMX(K)                                            PH2 3940
  521   CONTINUE                                                         PH2 3950
  522   IF(FS)9905,524,523                                               PH2 3960
  523   WS=U(K)**2+V(K)**2                                               PH2 3970
        ETH=ETH-AMMP*(AIX(K)+WS/2.0)                                     PH2 3980
        IF(AMMP/(TAU(I)*DY(J))-TOZONE)524,524,6901                       PH2 3990
 6901   REZ=1.0                                                          PH2 4000
  524   AMUR=AMMP*U(K)                                                   PH2 4010
        AMVR=AMMP*V(K)                                                   PH2 4020
  525   SIGMU=SIGMU-AMUR                                                 PH2 4030
        SIGMV=SIGMV-AMVR                                                 PH2 4040
  526   TIC=0.0                                                          PH2 4050
  527   IF(AMMP)528,529,529                                              PH2 4060
  528   DELER=AIX(K+1)+(U(K+1)**2+V(K+1)**2)/2.0                         PH2 4070
        GO TO 537                                                        PH2 4080
  529   IF(AMMY)530,531,531                                              PH2 4090
  530   DELER=DELEB                                                      PH2 4100
        GO TO 536                                                        PH2 4110
  531   IF(GAMC(J))532,533,533                                           PH2 4120
  532   DELER=SIGC(J)                                                    PH2 4130
        GO TO 536                                                        PH2 4140
  533   IF(AMPY)535,535,534                                              PH2 4150
  534   DELER=DELET                                                      PH2 4160
        GO TO 536                                                        PH2 4170
  535   DELER=AIX(K)+(U(K)**2+V(K)**2)/2.0                               PH2 4180
  536   TIC=1.0                                                          PH2 4190
  537   DELEK=DELEK-AMMP*DELER                                           PH2 4200
  538   IF(TIC)9907,539,550                                              PH2 4210
```

```
  550 WS=DELER                                                    PH2 4220
      GO TO 999                                                   PH2 4230
  539 WS=AIX(K)+(U(K)**2+V(K)**2)/2.0                             PH2 4240
  999 IF(DELM)998,543,540                                         PH2 4250
  998 IF(AMX(K)*1.E-6+DELM)9906,997,997                           PH2 4260
  997 DELM=0.0                                                    PH2 4270
      GO TO 543                                                   PH2 4280
C     ENK=TOTAL ENERGY OF CELL (K) + ENERGY THAT
C     HAS BEEN ADDED AND LOST.
  540 ENK=AMX(K)*WS+DELEK                                         PH2 4290
C     BY CONSERVING AXIAL MOMENTA, CALCULATE THE NEW
C     AXIAL VELOCITY COMPONENT FOR CELL K.
  541 U(K)=(SIGMU+AMX(K)*U(K))/DELM                               PH2 4300
C     BY CONSERVING RADIAL MOMENTA, CALCULATE THE NEW
C     RADIAL VELOCITY COMPONENT FOR CELL K.
  601 V(K)=(SIGMV+AMX(K)*V(K))/DELM                               PH2 4310
      IF(I-I1)603,6604,6604                                       PH2 4320
 6604 IF(U(K))6605,6606,6605                                      PH2 4330
 6605 NRC=1                                                       PH2 4340
 6606 IF(V(K))6607,6608,6607                                      PH2 4350
 6607 NRC=1                                                       PH2 4360
 6608 IF(AIX(K))6609,6610,6609                                    PH2 4370
 6609 NRC=1                                                       PH2 4380
 6610 CONTINUE                                                    PH2 4390
  603 WS=U(K)**2+V(K)**2                                          PH2 4400
C     BY CONSERVING BOTH TOTAL ENERGY AND
C     MOMENTA, CALCULATE THE NEW SPECIFIC
C     INTERNAL ENERGY FOR CELL K.
  542 AIX(K)=ENK/DELM-WS/2.0                                      PH2 4410
  543 AMX(K)=DELM                                                 PH2 4420
      IF(AMX(K))9900,2007,544                                     PH2 4430
 2007 AIX(K)=0.0                                                  PH2 4440
      U(K)=0.0                                                    PH2 4450
      V(K)=0.0                                                    PH2 4460
      P(K)=0.0                                                    PH2 4470
C     THE RIGHT VALUES OF CELL (K) BECOME THE LEFT
C     VALUES OF CELL (K+1).
  544 GAMC(J)=AMMP                                                PH2 4480
      FLEFT(J)=AMUR                                               PH2 4490
      YAMC(J)=AMVR                                                PH2 4500
      SIGC(J)=DELER                                               PH2 4510
C     THE TOP VALUES OF CELL(K) BECOME THE
C     BOTTOM VALUES FOR CELL (K+IMAX).
  545 AMMY=AMPY                                                   PH2 4520
      AMMU=AMUT                                                   PH2 4530
      AMMV=AMVT                                                   PH2 4540
      DELEB=DELET                                                 PH2 4550
  546 K=K+IMAX                                                    PH2 4560
      LL=K-IMAX                                                   PH2 4570
      IF(J(LL))6500,6600,6500                                     PH2 4580
```

```
6500 NRT=1                                                           PH2 4590
6600 IF(V(LL))6601,6602,6601                                         PH2 4600
6601 NRT=1                                                           PH2 4610
6602 IF(AIX(L))6611,547,6611                                         PH2 4620
6611 NRT=1                                                           PH2 4630
 547 CONTINUE                                                        PH2 4640
     I1=I1+NRC                                                       PH2 4650
     I2=I2+NRT                                                       PH2 4660
     IF(IMAX-I1)6700,6701,6702                                       PH2 4670
6700 I1=IMAX                                                         PH2 4680
6701 CONTINUE                                                        PH2 4690
6702 IF(JMAX-I2)6800,6801,6802                                       PH2 4700
6800 I2=JMAX                                                         PH2 4710
6801 CONTINUE                                                        PH2 4720
6802 GO TO 548                                                       PH2 4730
9901 NK=300                                                          PH2 4740
     GO TO 9999                                                      PH2 4750
9900 NK=302                                                          PH2 4760
     GO TO 9999                                                      PH2 4770
9903 NK=305                                                          PH2 4780
     GO TO 9999                                                      PH2 4790
9904 NK=506                                                          PH2 4800
     GO TO 9999                                                      PH2 4810
9905 NK=500                                                          PH2 4820
     GO TO 9999                                                      PH2 4830
9906 NK=513                                                          PH2 4840
     GO TO 9999                                                      PH2 4850
9911 NK=8833                                                         PH2 4860
     GO TO 9999                                                      PH2 4870
9908 NK= 17                                                          PH2 4880
     GO TO 9999                                                      PH2 4890
9909 NK= 22                                                          PH2 4900
     GO TO 9999                                                      PH2 4910
9910 NK= 47                                                          PH2 4920
     GO TO 9999                                                      PH2 4930
9907 NK=538                                                          PH2 4940
9999 NR=4                                                            PH2 4950
     CALL DUMP                                                       PH2 4960
 548 SUM=0.0                                                         PH2 4970
2005 DO 2001 I=1,I1                                                  PH2 4980
     K=I+1                                                           PH2 4990
     DO 2000 J=1,I2                                                  PH2 5000
     IF(AMX(K))2000,2000,2009                                        PH2 5010
C    IF ANY RHO (CELL DENSITY) IS LESS THAN TOZONE,
C    SET THE MASS TO ZERO, AND TALLY THE
C    MOMENTAS AND ENERGIES IN THE Z STORAGE, ALSO
C    CHECK FOR NEGATIVE INTERNAL ENERGIES, IF
C    WE FIND SOME, SET THEM TO ZERO AFTER
C    SUBTRACTING THEM FROM ETH..
2009 IF(AMX(K)/(TAU(I)*DY(J))-TOZONE)2010,2008,2008                  PH2 5020
```

```
 2010 WS=(U(K)**2+V(K)**2)/2.0                                    PH2 5030
      Z(100)=Z(100)+AMX(K)                                        PH2 5040
      WS=AMX(K)*(AIX(K)+WS)                                       PH2 5050
      Z(101)=Z(101)+WS                                            PH2 5060
      ETH=ETH-WS                                                  PH2 5070
      Z(102)=Z(102)+AMX(K)*U(K)                                   PH2 5080
      Z(103)=Z(103)+AMX(K)*V(K)                                   PH2 5090
      AMX(K)=0.0                                                  PH2 5100
      AIX(K)=0.0                                                  PH2 5110
      P(K)=0.0                                                    PH2 5120
      U(K)=0.0                                                    PH2 5130
      V(K)=0.0                                                    PH2 5140
      GO TO 2000                                                  PH2 5150
 2008 IF(AIX(K))2004,2000,2000                                    PH2 5160
 2004 SUM=SUM+AIX(K)*AMX(K)                                       PH2 5170
      AIX(K)=0.0                                                  PH2 5180
 2000 K=K+IMAX                                                    PH2 5190
 2001 CONTINUE                                                    PH2 5200
 2003 ETH=ETH-SUM                                                 PH2 5210
      Z(104)=Z(104)+SUM                                           PH2 5220
 8000 IF(REZ)8001,8001,8002
 8002 IF(REZFCT)8004,8004,8003
 8004 REZ=0.
      GO TO 8001
 8003 CALL REZONE
 8001 RETURN                                                      PH2 5260
      END                                                         PH2 5270
```

```
$IBFTC ES      LIST,DECK,REF
      SUBROUTINE ES                                              ES  0010
C                                                                ES  0760
C                                                                ES  0900
C     METALLIC EQUATION OF STATE, SEE
C     GA-3216 REPORT.
C                                                                ES  0980
   10 RHOW=AMX(K)/(TAU(I)*DY(J))                                 ES  0990
      ETA=RHOW/Z(115)                                            ES  1000
      VOW=1.0/ETA                                                ES  1010
   11 P1=AIX(K)*RHOW*Z(116)                                      ES  1020
   12 P2=(Z(115)*TAU(I)*DY(J))**2*AIX(K)                         ES  1030
   13 P3=AMX(K)**2*Z(117)                                        ES  1040
   14 P4=Z(118)/(P2/P3+1.0)*AIX(K)*RHOW                          ES  1050
   15 P5=Z(119)*(ETA-1.0)                                        ES  1060
   16 IF(ETA-1.0)50,100,100                                      ES  1070
   50 IF(VOW-Z(120))55,55,75                                     ES  1080
   55 IF(AIX(K)-Z(122))100,100,75                                ES  1090
   75 P7=Z(123)*(VOW-1.0)                                        ES  1100
      IF(P7-88.0)4002,4002,4003                                  ES  1110
 4003 P7=88.0                                                    ES  1120
 4002 CONTINUE                                                   ES  1130
      P8=EXP(P7)                                                 ES  1140
      P9=1.0/P8                                                  ES  1150
      P10=Z(124)*((VOW-1.0)**2)                                  ES  1160
      IF(P10-88.0)4000,4000,4001                                 ES  1170
 4001 P10=88.0                                                   ES  1180
 4000 CONTINUE                                                   ES  1190
      P11=EXP(P10)                                               ES  1200
      P12=1.0/P11                                                ES  1210
      P(K)=P1+(P4+P5*P9)*P12                                     ES  1220
      GO TO 119
  100 P6=Z(126)*((ETA-1.0)**2)                                   ES  1230
      P(K)=P1+P4+P5+P6                                           ES  1240
  119 IF(P(K))999,999,200                                        ES  1250
  200 WSGX=.5                                                    ES  1260
      GO TO 500                                                  ES  1270
  999 P(K)=0.0                                                   ES  1280
      WSGX=.5+Z(125)                                             ES  1290
      GO TO 500                                                  ES  1300
  500 RETURN                                                     ES  1310
      END                                                        ES  1320
$IBFTC ES      LIST,DECK,REF
      SUBROUTINE ES                                              ES  0010
C     ** POLYTROPIC EQUATION OF STATE **                         ES  0980
      P(K)=AMX(K)*AIX(K)/GAMX/(DY(J)*TAU(I))                     ES  0990
      WSGX=GMAX-1.0                                              ES  1000
      RETURN                                                     ES  1010
      END                                                        ES  1020
```

```
$IBFTC REZONE  LIST,DECK,REF
      SUBROUTINE REZONE
C
C     CONSERVE MOMENTUM AND TOTAL ENERGY, INCREASE
C     ALL LINEAR DIMENSIONS BY 2. (THUS 4 CELLS
C     IN THE OLD GRID ARE COMBINED INTO 1 FOR
C     THE NEW GRID.)
      NIMAX=IMAX/2
      NJMAX=JMAX/2
      DO 10 J=1,NJMAX
      K=(J-1)*NIMAX+2
      L=(J-1)*2*IMAX+2
      DO 11 I=1,NIMAX
      M=L+IMAX
   12 WSA=AMX(L)+AMX(M)+AMX(L+1)+AMX(M+1)
      WSB=AMX(L)*(U(L)**2+V(L)**2)+AMX(M)*(U(M)
     1**2+V(M)**2)+AMX(L+1)*(U(L+1)**2+V(L+1)**2)
     2+AMX(M+1)*(U(M+1)**2+V(M+1)**2)
      U(K)=(U(L)*AMX(L)+U(M)*AMX(M)+U(L+1)*AMX(L+1)+
     1U(M+1)*AMX(M+1))/WSA
      V(K)=(V(L)*AMX(L)+V(M)*AMX(M)+V(L+1)*AMX(L+1)+
     1V(M+1)*AMX(M+1))/WSA
      AIX(K)=AIX(L)*AMX(L)+AIX(M)*AMX(M)+AIX(L+1)*
     1AMX(L+1)+AMX(M+1)*AIX(M+1)
      AMX(K)=WSA
      WS=U(K)**2+V(K)**2
      E=AIX(K)+WSB/2.0
      AIX(K)=E/AMX(K)-.5*WS
      IF(K-2)14,14,13
C     SET CELL QUANTITIES OF OLD GRID TO ZERO.
   13 AMX(L)=0.0
      U(L)=0.0
      V(L)=0.0
      AIX(L)=0.0
      P(L)=0.0
      AMX(M)=0.0
      U(M)=0.0
      V(M)=0.0
      AIX(M)=0.0
      P(M)=0.0
      AMX(L+1)=0.0
      U(L+1)=0.0
      V(L+1)=0.0
      AIX(L+1)=0.0
      P(L+1)=0.0
      AMX(M+1)=0.0
      U(M+1)=0.0
      V(M+1)=0.0
      AIX(M+1)=0.0
      P(M+1)=0.0
```

```
   14 K=K+1                                                      REZ01410
      L=L+2                                                      REZ01420
   11 CONTINUE                                                   REZ01430
   10 CONTINUE                                                   REZ01440
C     CALCULATE NEW DY AND Y (JMAX OF THEM).
   18 DO 999 J=1,JMAX                                            REZ01450
      DY(J)=DY(J)*2.0                                            REZ01460
  999 CONTINUE                                                   REZ01470
      DO 99 J=1,JMAX                                             REZ01480
      Y(J)=Y(J-1)+DY(J)                                          REZ01490
   99 CONTINUE                                                   REZ01500
   16 DX(1)=2.0*DX(1)                                            REZ01510
      X(1)=DX(1)                                                 REZ01520
      WS=X(1)**2                                                 REZ01530
      TAU(1)=PIDY*WS                                             REZ01540
C     CALCULATE NEW DX AND X, AND TAU(IMAX OF THEM)
   17 DO 98 I=2,IMAX                                             REZ01550
      X(I)=X(I-1)+DX(I)                                          REZ01560
      DX(I)=DX(1)                                                REZ01570
      WSA=X(I)**2                                                REZ01580
      TAU(I)=PIDY*(WSA-WS)                                       REZ01590
      WS=WSA                                                     REZ01600
   98 CONTINUE                                                   REZ01610
      IMAX=NIMAX                                                 REZ01620
      JMAX=NJMAX                                                 REZ01630
C     PREPARE NOW TO SHUFFLE THE X ARRAYS SUCH
C     AS TO PRESERVE K=(J-1)*IMAX+I+1.
      DO 20 N=1,JMAX                                             REZ01640
      J=JMAX+1-N                                                 REZ01650
      K=(J-1)*IMAX+1+IMAX                                        REZ01660
      L=(J-1)*(IMAX+IMAX)+1+IMAX                                 REZ01670
      DO 21 I=1,IMAX                                             REZ01680
 1000 AMX(L)=AMX(K)                                              REZ01690
      AIX(L)=AIX(K)                                              REZ01700
      U(L)=U(K)                                                  REZ01710
      V(L)=V(K)                                                  REZ01720
      P(L)=P(K)                                                  REZ01730
      IF(J-1)1002,1002,1001                                      REZ01740
 1001 AMX(K)=0.0                                                 REZ01750
      AIX(K)=0.0                                                 REZ01760
      V(K)=0.0                                                   REZ01770
      U(K)=0.0                                                   REZ01780
      P(K)=0.0                                                   REZ01790
 1002 K=K-1                                                      REZ01800
      L=L-1                                                      REZ01810
   21 CONTINUE                                                   REZ01820
   20 CONTINUE                                                   REZ01830
      IMAX=NIMAX*2                                               REZ01840
      JMAX=NJMAX*2                                               REZ01850
      II=NIMAX+1
```

```
      JJ=NJMAX+1
C     ADD ON NEW MASS WITH DENSITY=Z(111) IN TARGET
      DO 50 I=1,NIMAX
      K=(JJ-1)*IMAX+I+1
      DO 60 J=JJ,JMAX
      AMX(K)=Z(111)*TAU(I)*DY(J)
   60 K=K+IMAX
   50 CONTINUE
      JJ=(Z(147)/2.+.2)
      JJ=JJ+1
      DO 61 I=I1,IMAX
      K=I+1+(JJ-1)*IMAX
      DO 62 J=JJ,JMAX
      AMX(K)=Z(111)*TAU(I)*DY(J)
   62 K=K+IMAX
   61 CONTINUE
C     RESET ACTIVE GRID MARKERS.
      JJ=JJ-1
      Z(147)=JJ
      I1=NIMAX+2
      I2=NJMAX+2
      WS=T+DTNA
      NK=NC+1
C     EDIT THE NEW GRID.
      WRITE (6,8004)WS,NK,DX(1)
      WRITE (6,8007)IMAX,(X(I),I=0,IMAX)
      WRITE (6,8008)JMAX,(Y(J),J=0,JMAX)
      WRITE (6,8009)IMAX,(DX(I),I=1,IMAX)
      WRITE (6,8010)JMAX,(DY(J),J=1,JMAX)
      WRITE (6,8011)IMAX,(TAU(I),I=1,IMAX)
      KMAX=IMAX*JMAX+1
      IMAXA=IMAX+1
      JMAXA=JMAX+1
      KMAXA=KMAX+1
      RETURN
 8004 FORMAT(1H ////22H PROBLEM REZONED AT T=1PE12.6,6X,5HCYCLEI4,6X,3HD
     1X=E12.6////)
 8007 FORMAT(1H /10H X(I) I=0,I2/(5F16.6))
 8008 FORMAT(1H /10H Y(J) J=0,I2/(5F16.6))
 8009 FORMAT(1H /11H DX(I) I=1,I2/(5F16.6))
 8010 FORMAT(1H /11H DY(J) J=1,I2/(5F16.6))
 8011 FORMAT(1H /13H AREA(I) I=1,I2/(F16.6,4F18.6))
      END
```

```
$IBFTC EDIT    LIST,DECK,REF
      SUBROUTINE EDIT                                                   EDIT0010
C                                                                       EDIT0990
C                                                                       EDIT1000
C     SENSE LITE (1) INDICATES LAST CYCLE OF THIS
C     RUN.
C     SENSE LITE (3) INDICATES FIRST CYCLE OF THIS
C     RUN.
  104 CALL SLITET(3,K000FX)                                             EDIT1040
       GO TO(106,108),K000FX                                            EDIT1050
  106 CALL SLITE (3)                                                    EDIT1060
      GO TO 126                                                         EDIT1070
  108 IF(CYCLE-CSTOP)110,122,122                                        EDIT1080
  110 IF(REZ)9901,112,124
  112 IF(AMOD(CYCLE,DUMPT7))114,124,114                                 EDIT1100
  114 IF(AMOD(CYCLE,PRINTL))120,126,120
  120 IF(AMOD(CYCLE,PRINTS))140,128,140                                 EDIT1150
C     NORMAL STOP ON THIS CYCLE.
  122 CALL SLITE (1)                                                    EDIT1160
C     DUMP ON TAPE 7.
  124 GO TO 1                                                           EDIT1170
  126 CALL SLITE (4)                                                    EDIT1180
  128 GO TO 6000                                                        EDIT1190
  130 GO TO 1000                                                        EDIT1200
  132 CALL SLITET(4,K000FX)                                             EDIT1210
       GO TO(134,136),K000FX                                            EDIT1220
  134 GO TO 5000                                                        EDIT1230
C     CHECK FOR ENERGY CHECK ERROR. WHERE
C     ECK= PERCENT ERROR/PER CYCLE.
C     ECK=(ETH-E)/ETH AT CYCLE N-(ETH-E)/ETH
C     AT CYCLE N-NPC ALL DIVIDED BY NPC. NOTE
C     NPC= NO. OF CYCLES BETWEEN ENERGY CHECK
  136 IF(ABS(ECK)-DMIN)140,140,9905                                     EDIT1240
  140 CALL SLITET(1,K000FX)                                             EDIT1250
       GO TO(142,144),K000FX                                            EDIT1260
 .142 REWIND  7
      CALL SLITE (1)                                                    EDIT1280
  144 GO TO 10000                                                       EDIT1290
C                                                                       EDIT1300
C                                                                       EDIT1310
C                     DUMP ON TAPE 7                                    EDIT1320
    1 IF(DUMPT7)30,3,3                                                  EDIT1330
    3 BACKSPACE  7
      REWIND 2
      REWIND 3
      WS=555.0                                                          EDIT1360
      WRITE ( 7)WS,CYCLE,N3
      WRITE ( 7)(Z(L),L=1,MZ)
    6 WRITE ( 7)(U(K),V(K),AMX(K),AIX(K),P(K),K=1,KMAXA)
    7 WRITE ( 7)X(0),(X(K),TAU(K),K=1,IMAX)
```

```
      WRITE ( 7)(Y(K),K=0,JMAX)
C     AGAIN, IF PROBLEM NO. IS NEGATIVE, WRITE
C     PARTICLE RECORDS ON TAPE 7.
      IF(PROB)16,16,4                                                  EDIT1420
   16 DO 18 I=1,N3                                                     EDIT1430
      IF(NI)150,148,150
  148 CONTINUE
      READ  ( 2)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,N4)
      GO TO 152
  150 CONTINUE
      READ  ( 3)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,N4)
  152 CONTINUE
      WRITE ( 7)(AM(N),XL(N),YL(N),IW1(N),IW2(N),N=2,N4)
   18 CONTINUE                                                         EDIT1460
    4 REWIND 2
      REWIND 3
      WS=666.0                                                         EDIT1480
      WRITE ( 7)WS,WS,WS
      WRITE (6,8120)NC                                                 EDIT1500
   30 GO TO 126                                                        EDIT1510
C                                                                      EDIT1520
C                                                                      EDIT1530
 6000 NK=12                                                            EDIT1540
C     TABS ARE TANGENT ALPHAS.
      TAB(1)=0.02                                                      EDIT1550
      TAB(2)=0.04                                                      EDIT1560
      TAB(3)=0.06                                                      EDIT1570
      TAB(4)=0.08                                                      EDIT1580
      TAB(5)=0.10                                                      EDIT1590
      TAB(6)=0.15                                                      EDIT1600
      TAB(7)=0.20                                                      EDIT1610
      TAB(8)=0.25                                                      EDIT1620
      TAB(9)=0.30                                                      EDIT1630
      TAB(10)=0.4                                                      EDIT1640
      TAB(11)=0.5                                                      EDIT1650
      TAB(12)=1.0                                                      EDIT1660
 6010 DO 6012 I=1,18                                                   EDIT1670
 6012 PR(I)=0.0                                                        EDIT1680
      NK1=NK+2                                                         EDIT1690
      DO 6014 I=1,NK1                                                  EDIT1700
      AMK(I)=0.0                                                       EDIT1710
      PK(I)=0.0                                                        EDIT1720
 6014 QK(I)=0.0                                                        EDIT1730
      DO 6028 K=2,KMAX                                                 EDIT1740
 6017 PR(1)=0.0                                                        EDIT1760
      PR(2)=0.0                                                        EDIT1770
      PR(4)=0.
C     CALCULATE KINETIC ENERGY IN CELL K.
      WSB=(U(K)**2+V(K)**2)*.5
 6019 IF(AMX(K))9917,6028,6020                                         EDIT1790
```

```
 6020  I=NK1                                                  EDIT1800
       IF(V(K))6026,6026,6022                                 EDIT1810
 6022  WSA=ABS(U(K))/V(K)                                     EDIT1820
       DO 6024 I=1,NK                                         EDIT1830
C      SEARCH FOR TAN ANGLE THAT VELOCITY VECTORS
C      MAKE.
       IF(TAB(I)-WSA)6024,6026,6026                           EDIT1840
 6024  CONTINUE                                               EDIT1850
       I=NK+1                                                 EDIT1860
 6026  WS=AMX(K)                                              EDIT1870
C      SUM UP MASS BETWEEN ANGLES.
 6027  AMK(I)=AMK(I)+AMX(K)                                   EDIT1880
C      SUM UP RADIAL MOMENTA IN THE ANGLES.
       PK(I)=PK(I)+U(K)*AMX(K)                                EDIT1890
C      SUM UP AXIAL MOMENTA IN THE ANGLES.
       QK(I)=QK(I)+V(K)*AMX(K)                                EDIT1900
C      SUM UP TOTAL INTERNAL ENERGY
       PR(5)=PR(5)+AIX(K)*AMX(K)                              EDIT1910
C      SUM UP TOTAL KINETIC ENERGY
       PR(6)=PR(6)+WSB*AMX(K)                                 EDIT1920
C      SUM UP TOTAL MASS
       PR(8)=PR(8)+AMX(K)                                     EDIT1930
 6028  CONTINUE                                               EDIT1940
       PR(3)=PR(1)+PR(2)                                      EDIT1950
       PR(7)=PR(5)+PR(6)                                      EDIT1960
       XNRG=PR(7)                                             EDIT1970
       PR(9)=PR(1)+PR(5)                                      EDIT1980
       PR(10)=PR(2)+PR(6)                                     EDIT1990
       PR(11)=PR(3)+PR(7)                                     EDIT2000
       PR(12)=PR(4)+PR(8)                                     EDIT2010
       WSA=(ETH-PR(11))/ETH                                   EDIT2020
       IF(CYCLE)9931,9931,9932
 9931  NPC=1
 9932  PR(18)=(WSA-DNN)/FLOAT(NPC)
       ECK=PR(18)                                             EDIT2040
       DNN=WSA                                                EDIT2050
C      RESET CYCLE COUNTER BETWEEN ENERGY CHECK.
       NPC=0                                                  EDIT2060
       SUM=0.0                                                EDIT2070
       DO 800 I=1,13                                          EDIT2080
       SUM=SUM+QK(I)                                          EDIT2090
  800  CONTINUE                                               EDIT2100
C      RADET= TOTAL POSITIVE AXIAL MOMENTUM IN GRID
       RADET=SUM                                              EDIT2110
  801  SUM=0.0                                                EDIT2120
       DO 810 K=2,KMAX                                        EDIT2130
       IF(AMX(K))810,810,802                                  EDIT2140
  802  IF(U(K))810,810,803                                    EDIT2150
  803  SUM=SUM+AMX(K)*U(K)                                    EDIT2160
  810  CONTINUE                                               EDIT2170
```

```
C      RADEF= TOTAL POSITIVE RADIAL MOMENTUM IN CELL.
       RADEF=SUM
       SUM=0.0
       JJ=Z(147)
       DO 811 I=1,IMAX
       K=I+1
       DO 813 J=1,JJ
       IF(AMX(K))813,813,814
  814  IF(U(K))813,813,815
  815  SUM=SUM+W(K)*AMX(K)
  813  K=K+IMAX
  811  CONTINUE
C      RADEB= TOTAL POSITIVE RADIAL MOMENTUM BELOW
C      INITIAL TARGET-PROJECTILE INTERFACE.
       RADEB=SUM
       PR(19)=0.0
       DO 6029 I=1,NK
 6029  PR(I+19)=PR(I+18)+AMX(I)
       PR(NK+20)=0.0
       PR(NK+21)=0.0
       WRITE (6,8116)PROB,NC,T,CYCLE,TRAD,DTRAD,NR,N1,N2,N3,N4
       WRITE (6,8117)(PR(I),I=1,5)
       WRITE (6,8118)(PR(I),I=6,12)
       WRITE (6,8119)RADEB,RADEF,RADET,UMAX,ETH,ECK
       WRITE (6,9040)MIC,MI1,I1,I2,I3,I4
       WRITE (6,8124)(I,AMX(I),PR(I+19),PK(I),QK(I),I=1,NK1)
 6090  GO TO 130
C**** END OF   S P   SUBROUTINE *******************************************
C
C
C**** SUBROUTINE PLOT ******* ***********************************************
 1000  GO TO 1030
 1030  WRITE (6,8116)PROB,NC,T,CYCLE,TRAD,DTRAD,NR,N1,N2,N3,N4
       JMAX=JMAX
       WRITE (6,8307)X1,X2,XMAX,Y1,Y2,Y(JMAX)
       M=1
       IF(JMAX-52)1034,1036,1036
 1034  M=IABS(51-JMAX)/2
 1036  DO 1040 I=1,M
       WRITE (6,6308)
 1040  CONTINUE
 1044  J=J2
 1100  K=(J-1)*IMAX+1
 1105  DO 1130 I=1,I1
       K=K+1
C      REPLACE 6000000000000 BY-17179869184
 1126  PR(I)=(-ABS(-17179869184))
 1150  IF(AMX(K))9917,1156,1160
C        X PARTICLE ONLY
C      REPLACE 6700000000     BY    922746880
```

```
 1160 PR(I)=OR(PR(I),  ABS(   922746880) )                              EDIT2650
      GO TO 1180                                                        EDIT2660
C     REPLACE 6000000000   BY   805306368                               EDIT2670
 1166 PR(I)=OR(PR(I),  ABS(   805306368) )                              EDIT2680
 1180 CONTINUE                                                          EDIT2690
 1200 IF(MOD(J,5))1210,1204,1210                                        EDIT2700
 1204 IF(DY(J)-DY(J-1))1206,1208,1206                                   EDIT2710
 1206 WRITE (6,8211)DY(J),J,(PR(I),I=1,I1)                              EDIT2720
      GO TO 1224                                                        EDIT2730
 1208 WRITE (6,8201)J,(PR(I),I=1,I1)                                    EDIT2740
      GO TO 1224                                                        EDIT2750
 1210 IF(DY(J)-DY(J-1))1212,1214,1212                                   EDIT2760
 1212 WRITE (6,8222)DY(J),(PR(I),I=1,I1)                                EDIT2770
      GO TO 1224                                                        EDIT2780
 1214 WRITE (6,8202)(PR(I),I=1,I1)                                      EDIT2790
 1224 J=J-1                                                             EDIT2800
 1226 IF(J)1230,1230,1100                                               EDIT2810
C     REPLACE 604000000000 BY-17716740096                               EDIT2820
 1230 PR(1)=(-ABS(-17716740096))                                        EDIT2830
      WRITE (6,8201)J,(PR(1),I=1,I1)                                    EDIT2840
      WRITE (6,8302)(I,I=0,IMAX,5)                                      EDIT2850
 1240 GO TO 132                                                         EDIT2860
C**** END OF  PLOT  SUBROUTINE ******************************************EDIT2870
C                                                                       EDIT2880
C                                                                       EDIT2890
C**** SUBROUTINE L P ****************************************************EDIT2900
 5000 WRITE (6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4            EDIT2910
 5004 DO 5050 I=1,I1                                                    EDIT2920
      CALL SLITE (4)                                                    EDIT2930
      J=I2+1                                                            EDIT2940
      K=I2*IMAX+1+I                                                     EDIT2950
      DO 5046 L=1,I2                                                    EDIT2960
      J=J-1                                                             EDIT2970
      K=K-IMAX                                                          EDIT2980
 5012 IF(AMX(K))9917,5046,5014                                          EDIT2990
 5014 CALL SLITET(4,K000FX)                                             EDIT3000
       GO TO(5016,5019),K000FX                                          EDIT3010
 5016 WRITE (6,8135)I,X(I),DX(I)                                        EDIT3020
C     WS= DENSITY OF CELL(K) IN GRAMS/CM. CUBED.
 5019 WS=AMX(K)/(TAU(I)*DY(J))                                          EDIT3030
C     WSA= COMPRESSION = RHO/RHO NOT.
      WSA=WS/Z(111)                                                     EDIT3040
C     WSC= PRESSURE CONVERTED TO MEGABARS.
      WSC=P(K)*1.E+4                                                    EDIT3050
C     FIRST COLUMN= (J) THE ROW NO.
C     SECOND COLUMN= RADIAL VELOCITY CM./SHAKE
C     THIRD COLUMN= AXIAL VELOCITY CM./SHAKE
C     FOURTH COLUMN= F/A = PRESSURE IN MEGABARS
C     FIFTH COLUMN = AMX = MASS IN GRAMS.
C     SIXTH COLUMN = RHO = DENSITY IN GRAMS/CC.
```

```
C      SEVENTH COLUMN = AIX = SPECIFIC INTERNAL ENERGY JERKS/GM.
C      EIGHT COLUMN = COMPRESSION = RHO/RHO NOT
C      NINTH COLUMN = Z VALUE (CM.) OF TOP OF CELL)
 50180WRITE (6,8108)J,U(K),V(K),WSC,AMX(K),                   WS,AIX(K),
     1WSA,Y(J)
 5046 CONTINUE
 5050 CONTINUE
      GO TO 136
C**** END OF   L P  SUBROUTINE ****************************************
C
C
C              ERROR
 9901 NK=110
      GO TO 9999
C              ENERGY CHECK
 9905 NK=136
      GO TO 9999
C              NEGATIVE MASS
 9917 NK=6015
      GO TO 9999
 9920 NK=904
      GO TO 9999
 9921 NK=912
      GO TO 9999
 9922 NK=918
      GO TO 9999
 9923 NK=922
      GO TO 9999
 9924 NK=926
 9999 NR=6
      CALL DUMP
10000 RETURN
C
C              FORMATS
 8108 FORMAT(I3,1X,1P2E14.6,3E15.6,E14.6,E15.6,E14.6)
 81160FORMAT(8H1PROBLEM6X,5HCYCLE9X,4HTIME13X,2HDT13X,4HTRAD11X,5HDTRAD1
     12X,2HNR6X,2HN14X,2HN24X,2HN34X,2HN4/(F7.1,I11,2X,1P4E16.7,I10,2X,4
     2I6))
 81170FORMAT(1H0//17X2HAI16X,2HAK14X,5HAI+AK15X,2HAM/4H DOT3X,1P4E18.7/3
     1H  X4X,4E18.7)
 81180FORMAT(12X,13H-------------5X,13H-------------5X,13H-------------5
     1X,13H-------------/7H TOTALS1P4E18.7)
 81190FORMAT(2H0 //16X,5HRADEB13X,5HRADER13X,5HRADET12X,7HMAX VEL13X,3HT
     1HE12X,9HREL ERROR/7X,1P6E18.7////)
 8120 FORMAT(1H0//21H TAPE 7 DUMP ON CYCLEI5////)
 81240FORMAT(3H  K12X,5HAM(K)11X,9HSUM AM(K)11X,4HP(K)13X,4HQ(K)/(I3,4X,
     11P4E18.7))
 8131 FORMAT(1H ///11H DY(J) J=1,I2//(10F12.3))
 8133 FORMAT(1H ///11H  Y(J) J=0,I2//(10F12.3))
 81350FORMAT(1H ///4H I =I3,6X,6HX(I) =F12.3,6X,7HDX(I) =F12.3//3H  J8X,
```

```
    11HX13X,1HY13X,3HF/A12X,3HAMX12X,3HRHO11X,3HAIX12X,4HCOMP11X,2H Y/)EDIT3530
 8201 FORMAT(I10,2H I54A2)                                              EDIT3540
 8202 FORMAT(10X,2H I54A2)                                              EDIT3550
 8211 FORMAT(F7.1,I3,2H I54A2)                                          EDIT3560
 8222 FORMAT(F7.1,3X,2H I54A2)                                          EDIT3570
 8302 FORMAT(I12,10I10)                                                 EDIT3580
 83070FORMAT(5H X1 =1PE12.6,3X,4HX2 =E12.6,3X,6HXMAX =E12.6,6X,4HY1 =E12EDIT3590
    1.6,3X,4HY2 =E12.6,3X,6HYMAX =E12.6)                                EDIT3600
 8308 FORMAT(1H /)                                                      EDIT3610
 9040 FORMAT(1H / 616)                                                  EDIT3620
      END                                                               EDIT3630
```

REFERENCES

1. Evans, M. W. and F. H. Harlow, "The Particle-in-Cell Method for Hydrodynamics Calculations," Los Alamos Scientific Laboratory Report LA-2139, November 1957.

2. Harlow, F. H., "Two-Dimensional Hydrodynamic Calculations," Los Alamos Scientific Laboratory, Report LA-2301, September 1959.

3. Johnson, W. E., "Computer Development to Improve SHELL Code,", General Atomic report GA-4673, Contract AF29(601)-6028, October 1963.

4. Walsh, J. M., W. E. Johnson, J. K. Dienes, J. H. Tillotson, and D. R. Yates, "Summary Report on the Theory of Hypervelocity Impact," General Atomic Report GA-5119, Contract DA-04-495-AMC-116(X), March 1964.

5. Tillotson, J. H., "Metallic Equations of State for Hypervelocity Impact," General Atomic report GA-3216, Contract AF29(601)-4759, July 1962.

6. Rich, M., "A Method for Eulerian Fluid Dynamics," Los Alamos Scientific Laboratory Report LAMS-2826, March 1963.